ПРИОРИТЕТНЫЙ НАЦИОНАЛЬНЫЙ ПРОЕКТ «ОБРАЗОВАНИЕ»
РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Д.С. КУЛЯБОВ, А.В. КОРОЛЬКОВА

АРХИТЕКТУРА И ПРИНЦИПЫ ПОСТРОЕНИЯ СОВРЕМЕННЫХ СЕТЕЙ И СИСТЕМ ТЕЛЕКОММУНИКАЦИЙ

Учебное пособие

Москва

2008

Инновационная образовательная программа
Российского университета дружбы народов

«Создание комплекса инновационных образовательных программ и формирование инновационной образовательной среды, позволяющих эффективно реализовывать государственные интересы РФ через систему экспорта образовательных услуг»

Экспертное заключение –

доктор технических наук, профессор *С.Н. Степанов*

Кулябов Д.С., Королькова А.В.

Архитектура и принципы построения современных сетей и систем телекоммуникаций: Учеб. пособие. – М.: РУДН, 2008. – 309 с.: ил.

В учебном пособии рассматриваются архитектура и принципы построения современных сетей и систем телекоммуникаций, основные протоколы и технологии.

Учебное пособие предназначено для студентов направлений 550200 «Автоматизация и управление», 511200 «Математика, прикладная математика», 510400 «Физика», 521500 «Менеджмент», 521600 «Экономика», 060800 «Экономика и управление на предприятии (по отраслям производства)».

Учебное пособие выполнено в рамках инновационной образовательной программы Российского университета дружбы народов, направление «Комплекс экспортоориентированных инновационных образовательных программ по приоритетным направлениям науки и технологий», и входит в состав учебно-методического комплекса, включающего описание курса, программу и электронный учебник.

Оглавление

Введение

Объединение компьютеров в сеть изменило парадигму обработки информации. Главное назначение сетей на сегодняшний момент — передача информации. В настоящее время существует множество сетевых технологий со своими особенностями, критериями применимости и пр. Оценить современное состояние сферы телекоммуникаций невозможно без понимания генезиса, принципов функционирования существующих систем и сетей связи.

Учебное пособие предназначено для студентов, обучающихся по программе дополнительного образования «Информационно-телекоммуникационные системы». В рамках инновационной образовательной программы, реализованной в РУДН в 2008–2009 гг. на кафедре систем телекоммуникаций, разработан одноимённый учебно-методический комплекс (УМК), в состав которого входит электронный учебник. Программа дополнительного образования является авторской и включает в себя набор последовательно взаимоувязанных специальных дисциплин.

На программе могут обучаться студенты, не имеющие специального образования, например, обучающиеся по направлениям «Автоматизация и управление», «Математика, прикладная математика», «Физика», «Менеджмент», «Экономика», «Экономика и управление на предприятии (по отраслям производства)». Курс является составляющей модуля программы дополнительной профессиональной подготовки «Основы управления инфокоммуникациями», которая включает также курсы: «Архитектура и принципы построения современных сетей и систем телекоммуникаций», «Введение в управление инфокоммуникациями», «Корпоративные информационные системы».

Учебное пособие логически разбито на две части. Первая часть изучает протоколы компьютерных сетей согласно структуре эталонной модели ISO/OSI. Основное внимание уделяется двум стекам протоколов — Ethernet и TCP/IP. Вторая часть рассматривает современное состояние сетей телекоммуникаций.

В первой главе даются и объясняются такие базовые понятия систем телекоммуникаций, как протокол, интерфейс, служба. Даётся обзор существующих сетей связи, сетевых сервисов. Рассматриваются структура и основные аспекты деятельности стандартизирующих организаций.

Во второй главе освещаются общие принципы построения модели взаимодействия открытых систем (ISO/OSI), иерархия протоколов различных стеков протоколов по отношению к модели ISO/OSI.

В третьей главе рассматриваются методы и технологии физического уровня модели ISO/OSI. В частности, даётся обзор возможных сред передачи (в том числе и стандарты кабельной системы), методов кодирования сигнала.

В четвёртой главе изучаются методы и протоколы доступа к среде, а также технологии сетей (Ethernet, Fast Ethernet, Gigabit Ethernet, Wireless Networks, WiMAX и т.д.). Упор делается на стандарты IEEE 802.x.

В пятой главе рассматриваются протоколы межсетевого уровня стека протоколов TCP/IP. Особое внимание уделяется протоколу IP: приведены формат кадра IP, схемы и правила IP-адресации (IPv4 и IPv6). В этой же главе отдельным пунктом представлена проблема маршрутизации: классификация алгоритмов маршрутизации, протоколы статической (iproute2, click) и динамической (RIP, OSPF, BGP) маршрутизации, сфера их применения. Кратко рассматриваются другие протоколы межсетевого уровня стека протоколов TCP/IP, их назначение.

Шестая глава посвящена протоколам транспортного уровня стека протоколов TCP/IP: TCP, UDP, DCCP, SCTP.

В седьмой главе описываются протоколы верхних уровней стека TCP/IP, а именно два протокола сеансового уровня DNS и ENUM, ответственные за адресацию.

В восьмой главе вводятся базовые понятия QoS, рассматриваются специальные решения обеспечения QoS, такие как организация виртуальных каналов в ATM, а также решения для IP-сетей и Ethernet — организация виртуальных каналов при помощи меток (MPLS), разбиение трафика на классы в соответствии с приоритетами каждого типа трафика и определение политик обслуживания этих классов трафика (DiffServ и IntServ). В связи с этим рассматриваются инструменты классификации и маркировки пакетов, а также механизмы планирования и выравнивания трафика.

Девятая глава посвящена мультисервисным сетям. Также в ней описываются основные подходы к построению сетей следующего поколения (NGN). В исторической ретроспективе рассматриваются два основных подхода к построению конвергентных сетей — Softswitch и IMS, даётся сравнительный обзор их концепций, освещаются архитектурные особенности и основные протоколы обоих подходов.

Глава 1. Общая характеристика проблемной области. Базовые понятия в области систем и сетей телекоммуникаций. Стандартизирующие организации

1.1. Базовые понятия в области систем и сетей телекоммуникаций

1.1.1. Сеть связи. Режимы передачи. Технологии коммутации

Сеть связи — это совокупность линий связи и промежуточного оборудования/промежуточных узлов, терминалов/оконечных узлов, предназначенных для передачи информации от отправителя до получателя с заданными параметрами качества обслуживания.

Линия связи представляет собой совокупность физической среды распространения сигналов и оборудования, формирующих специализированные каналы, имеющие определённые стандартные показатели: полосу частот, скорость передачи и т.п.

Каналы связи могут быть *непрерывными (аналоговыми)* и *дискретными (цифровыми)*. Также каналы связи различаются по направленности передачи. Выделяют три типа передачи информации:

- *симплексная передача (Simplex Transmission)* — передача данных в одном, предварительно определённом направлении;
- *полудуплексная передача (Half-Duplex Transmission)* — передача данных, при которой данные пересылаются в обоих направлениях, но только в одном направлении в каждый момент времени;
- *дуплексная передача (Duplex Transmission)* — передача данных, при которой данные пересылаются одновременно в обоих направлениях.

Обмен информацией между узлами сети обеспечивается с помощью *технологий коммутации*:

- *коммутация каналов (Circuit Switching)* — режим передачи, при котором формируется составной канал (соединение) через несколько транзитных узлов из нескольких последовательно «соединённых» каналов на время передачи информации (до разъединения соединения);
- *коммутация сообщений (Messege Switching)* — режим передачи, включающий приём, хранение, выбор исходящего направления и дальнейшую передачу сообщений без нарушения их целостности;
- *коммутация пакетов (Packet Switching)* — режим передачи сообщений, при котором сообщения разбиваются на пакеты ограниченного размера, причём канал передачи занят только во время передачи пакета и освобождается после её завершения;
- *коммутация ячеек (Cell Switching)* — режим передачи пакетов фиксированного размера.

1.1.2. Понятие протокола. Иерархия протоколов. Интерфейсы и сервисы

В широком смысле под протоколом понимается правило взаимодействия двух сущностей. Сетевой протокол определяет набор правил, позволяющих осуществлять соединение и обмен информацией между двумя элементами (узлами) сети.

Большинство протоколов строится как иерархический набор *уровней (Layers)*, каждый последующий из которых вводится над предыдущим (рис. 1.1). Нижележащий уровень предоставляет некоторый набор услуг (сервисов) для вышележащего, скрывая детали реализации предоставляемой услуги.

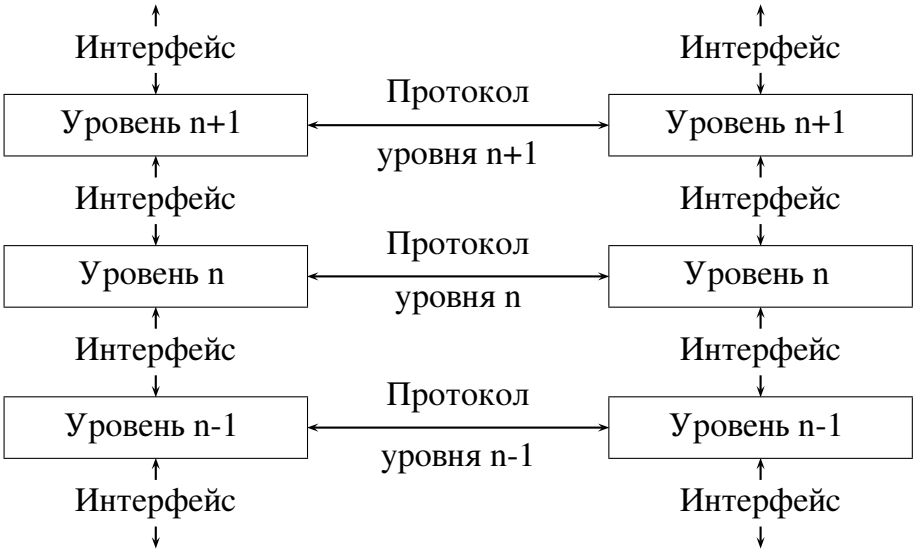


Рис. 1.1. Уровни протоколов

Взаимодействие производится между уровнем n одного узла и уровнем n другого. Используемые правила и соглашения называются *протоколом уровня* n. Между парой смежных уровней находится *интерфейс*, определяющий набор сервисов, предоставляемых нижележащим уровнем вышележащему.

Активный элемент каждого уровня называется *сущностью (Entity)*. Сущности одного уровня на разных узлах называются *одноранговыми сущностями*. Сущности уровня n (*поставщики услуг*) реализуют услуги, используемые уровнем $n+1$ (*потребители услуг*). Для предоставления этих услуг уровень n может использовать услуги уровня $n-1$.

Сервис, или услуга (Service), представляет собой набор примитивов, которые предоставляются вышележащему уровню нижележащим. Сервис определяет, какие именно операции уровень будет выполнять от лица своих пользователей, но никак не оговаривает, как должны реализовываться эти операции. Сервис описывает интерфейс между двумя уровнями, в котором нижележащий уровень является поставщиком услуги, а вышележащий — её потребителем.

Протокол определяет набор правил, описывающих формат и назначение пакетов, которыми обмениваются одноранговые сущности внутри уровня. Сущности используют протокол для реализации определений их сервисов. Протоколы

могут меняться, но предоставляемые услуги должны оставаться неизменными.

Услуги доступны через *точки доступа к услуге (Service Access Point, SAP)*. Чтобы два уровня могли обмениваться информацией, необходима договорённость о наборе правил используемого интерфейса. Сущность уровня $n + 1$ передаёт *элемент данных интерфейса (Interface Data Unit, IDU)*, состоящий из *элемента данных услуги (Service Data Unit, SDU)* и некоторой *управляющей информации (Interface Control Information, ICI)*, сущности с номером n через точку SAP. Для передачи SDU сущности уровня n может понадобиться разбить его на несколько фрагментов и послать их в виде отдельных *элементов данных протокола (Protocol Data Unit, PDU)* или *пакетов*.

По типу установления соединения протоколы можно разделить на два типа:

— протоколы с *установлением соединения (Connection Oriented)* — перед обменом данными отправитель и получатель должны сначала установить соединение и, возможно, выбрать некоторые параметры протокола, а после завершения сеанса они должны разорвать соединение;

— протоколы *без предварительного установления соединения (Connectionless)*, или *датаграммные (дейтаграммные)* протоколы — передача данных осуществляется, не дожидаясь установления соединения.

Используемый системой список протоколов называется *стеком протоколов*. Набор уровней и протоколов называется *архитектурой сети*.

1.2. Принципы классификации сетей телекоммуникаций

Сети телекоммуникаций можно классифицировать[1] по нескольким параметрам:

1) по размеру сети:
   — *локальные сети (Local Area Network, LAN)* — сети здания или организации;
   — *региональные сети (Metropolitan Area Network, MAN)* — сети уровня города или региона;
   — *глобальные сети (Wide Area Network, WAN)* — сети, охватывающие большие территории и включающие в себя десятки и сотни тысяч компьютеров;
2) по типу коммутации:
   — сети с коммутацией пакетов (например, TCP/IP, IPX/SPX, ATM, сети сотовой связи 3G);
   — сети с коммутацией каналов (например, ТфОП, сети сотовой связи 1G и 2G);
   — смешанные (например, сети сотовой связи 2,5G);
3) по установлению виртуального канала:
   — с установлением виртуального канала (например, сети X.25, Frame Relay, ATM, ТфОП);
   — без установления виртуального канала (например, TCP/IP, IPX/SPX);

[1] Заметим, что мы строим классификацию, а не таксономию.

4) по используемому стеку протоколов;
5) по количеству используемых стеков протоколов:
   - монопротокольные сети;
   - мультипротокольные сети (например, IP over ATM, IP over SDH/SONET);
6) по спектру оказываемых услуг:
   - моносервисные сети (передача данных, передача голоса);
   - мультисервисные сети;
7) по типу передаваемой информации:
   - сети передачи данных;
   - сети передачи голоса;
   - сети передачи видео;
8) по наличию сигнализации:
   - сети с выделенной сигнализацией (SS7);
   - сети без выделенной сигнализации (TCP/IP);
9) по топологии сети:
   - сети с топологией шина;
   - сети с топологией кольцо;
   - сети с топологией звезда;
   - сети со смешанной топологией;
10) по среде передачи:
   - проводные сети:
     - связь осуществляется по медному кабелю;
     - связь осуществляется по оптоволокну;
   - беспроводные сети.

1.3. Стандартизирующие организации

Организации в международной системе стандартизации можно разделить следующим образом:

- официальные международные организации стандартизации:
  - *Международная организация по стандартизации (International Organization for Standardization, ISO)*

    Создана в 1946 г., включает в себя национальные организации стандартизации из 157 стран мира, в частности, ANSI (США), Федеральное агентство по техническому регулированию и метрологии (Россия), BSI (Великобритания), AFNOR (Франция) и др., обладает полномочиями для координирования на международном уровне разработки различных промышленных стандартов и принятия их в качестве международных стандартов.

  - *Международный союз электросвязи, МСЭ (International Telecommunication Union, ITU[1])*

    Занимается стандартизацией международных средств связи и состоит из трёх основных секторов:

    * сектор стандартизации телекоммуникаций (ITU-T[2]) — занимается вопросами, связанными с телефонными системами и системами передачи данных[3];
    * сектор радиосвязи (ITU-R) — распределяет радиочастоты между конкурирующими компаниями, решает спорные вопросы в данной области;
    * сектор развития (ITU-D) — занимается вопросами стратегии и политики развития систем электросвязи;

- региональные организации стандартизации:
  - *Европейский институт стандартизации в области телекоммуникаций (European Telecommunications Standards Institute, ETSI)*

    Создан в 1988 г. Отвечает за стандартизацию информационных и телекоммуникационных технологий в пределах Европы.
  - *Центр сетевых информационных технологий Азиатско-Тихоокеанского региона (Asia Pacific Network Information Centre, APNIC)*

    Отвечает за распределение сетевых ресурсов в Азиатско-тихоокеанском регионе;
- национальные организации стандартизации:
  - *Федеральное агентство по техническому регулированию и метрологии (Россия)*;
  - *Американский институт национальных стандартов (American National Standards Institute, ANSI)*;
  - и др.;
- промышленные консорциумы:
  - *Сообщество инженеров по электротехнике и электронике (Institute of Electrical and Electronic Engineers, IEEE)*

    Целью данной организации является продвижение теоретических и прикладных достижений электротехнической и электронной индустрии.
  - *Рабочая группа по проектированию Интернет-технологий (Internet Engineering Task Force, IETF)*

    IETF представляет собой сообщество разработчиков, операторов, изготовителей и исследователей в области сетевых технологий. В основе Интернет-стандартизации лежит технология издания и поддержания RFC-документов — спецификаций, разработанных различными организациями и рабочими группами IETF.

---

[1] ITU также принято переводить как Международный телекоммуникационный союз.

[2] С 1956 по 1993 г. ITU-T именовался CCITT (Comité Consultatif International Télégraphique et Téléphonique) — Консультативный комитет по международной телефонной и телеграфной связи.

[3] Рекомендации ITU-T часто становятся международными стандартами, хотя правительство любой страны может принять или проигнорировать их.

– *Интернет-сообщество (Internet Society, ISOC)*
  ISOC представляет собой ассоциацию экспертов, отвечающих за разработку стандартов технологий сети Интернет.
– *Консорциум, специализирующийся в области разработки и развития стандартов WWW-технологий (World Wide Web Consortium, W3C).*

Глава 2. Модель ISO/OSI. Иерархия протоколов различных стеков относительно модели ISO/OSI

2.1. Обзор эталонной модели OSI

В начале 1980-х гг. ряд международных организаций по стандартизации (ISO, ITU-T) разработали *эталонную модель взаимодействия открытых систем (International Standards Organization / Open System Interconnection Reference Model, ISO/OSI)*. Модель ISO/OSI чётко определяет уровни взаимодействия систем, стандартизует имена уровней и указывает услуги и функции каждого уровня (рис. 2.1).

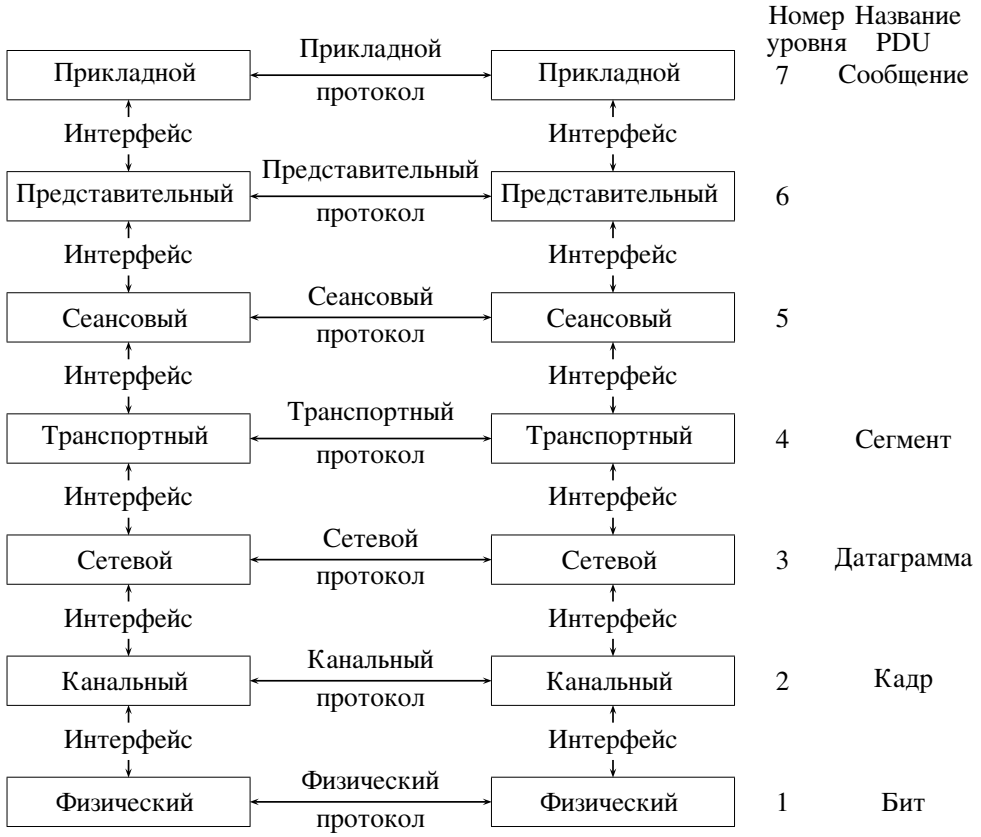


Рис. 2.1. Эталонная модель ISO/OSI

2.1.1. Принципы построения эталонной модели ISO/OSI

Эталонная модель ISO/OSI базируется на следующих принципах:

1) уровень должен создаваться по мере необходимости выделения отдельного уровня абстракции;
2) каждый уровень должен выполнять строго определённую функцию;
3) функции для каждого уровня должны выбираться с учётом создания стандартизованных международных протоколов;

4) границы между уровнями должны выбираться так, чтобы поток данных между интерфейсами был минимальным;
5) количество уровней должно быть достаточно большим, чтобы различные функции не объединялись в одном уровне без необходимости, но не слишком высоким, чтобы архитектура не становилась громоздкой.

2.1.2. Уровни в модели OSI

Одним из важнейших принципов OSI является то, что сетевые системы взаимодействуют друг с другом на одинаковых уровнях модели. Дадим краткое описание уровней модели OSI (рис. 2.1).

2.1.2.1. Уровень 1: Физический уровень

Физический уровень (Physical Layer) обеспечивает передачу битовых потоков без каких-либо изменений между логическими объектами уровня звена данных по физическим соединениям.

На данном уровне определяются базовые механизмы кодирования и декодирования двоичных данных в физическом носителе, а также специфицируются соединители, но не сама среда. Среда, согласно эталонной модели, рассматривается как нечто, лежащее ниже физического уровня. Битовый поток в носителе должен быть независим от типа среды.

Физический уровень предоставляет канальному уровню следующие услуги и элементы услуг:

— физические соединения;
— физические сервисные блоки данных;
— физические оконечные пункты соединения;
— осуществляет идентификацию канала данных;
— осуществляет упорядочение;
— осуществляет оповещение об ошибках;
— определяет параметры качества услуги.

На физическом уровне выполняются следующие функции:

— активизация и деактивизация физического соединения;
— передача физических сервисных блоков данных;
— административное управление физическим уровнем.

2.1.2.2. Уровень 2: Канальный уровень

Канальный уровень (Data Link Layer) также носит названия *уровень управления передачей данных (Data Link Control, DLC)* или *уровень звена данных*.

Канальный уровень обеспечивает функциональные и процедурные средства для установления, поддержания и разрыва соединений канального уровня между сетевыми логическими объектами и для передачи сервисных блоков данных этого уровня. Соединение канального уровня строится на основе одного или нескольких физических соединений.

Канальный уровень обнаруживает и по возможности исправляет ошибки, которые могут возникнуть на физическом уровне. Кроме того, канальный уровень обеспечивает для сетевого уровня возможность управлять подключением каналов данных на физическом уровне. Единицу информации на канальном уровне называют *кадром (Frame)*.

Канальный уровень предоставляет следующие услуги или элементы услуг сетевому уровню:
- соединение канального уровня;
- сервисные блоки данных канального уровня;
- идентификаторы оконечного пункта соединения канального уровня;
- осуществляет упорядочение блоков данных;
- осуществляет оповещение об ошибках;
- управляет потоком данных;
- определяет параметры качества услуги.

На канальном уровне выполняются следующие функции:
- установление и разрыв соединения канального уровня;
- отображение сервисных блоков данных канального уровня;
- расщепление соединения канального уровня;
- разграничение и синхронизация;
- упорядочение блоков данных;
- обнаружение ошибок;
- восстановление при ошибках;
- управление потоком данных;
- идентификация и обмен параметрами;
- управление переключением каналов данных;
- административное управление канальным уровнем.

2.1.2.3. Уровень 3: Сетевой уровень

Сетевой уровень (Network Layer) предоставляет средства установления, поддержания и разрыва сетевого соединения, а также функциональные и процедурные средства для обмена по сетевому соединению сетевыми сервисными блоками данных между транспортными логическими объектами.

Сетевой уровень обеспечивает транспортным логическим объектам независимость от функций маршрутизации и ретрансляции, связанных с процессами установления и функционирования данного сетевого соединения.

Все функции ретрансляции и расширенные протоколы последовательного переноса данных, которые предназначены для поддержания сетевых услуг между оконечными открытыми системами, функционируют ниже транспортного уровня. Единицу информации на сетевом уровне называют *датаграммой или дейтаграммой (Datagram)*.

Основной услугой сетевого уровня является обеспечение передачи данных без каких-либо изменений между транспортными логическими объектами, т.е. структура и содержание данных, предоставляемых для передачи, определяется уровнями, расположенными выше сетевого.

Услуги, предоставляемые на каждом из концов сетевого соединения, одинаковы и в том случае, когда сетевое соединение проходит через несколько подсетей, каждая из которых предоставляет различные услуги.

Сетевой уровень предоставляет следующие услуги:
- сетевые адреса;
- сетевые соединения;
- сетевые идентификаторы оконечных пунктов соединения;
- осуществляет передачу сетевых сервисных блоков данных;
- определяет параметры качества услуги;
- оповещает об ошибках;

— упорядочивает блоки данных;
— управляет потоком данных;
— осуществляет передачу срочных сетевых сервисных блоков данных;
— осуществляет сброс;
— осуществляет разрыв сетевого соединения.

Некоторые из этих услуг являются необязательными, т.е.:

— пользователь должен запросить услугу;
— поставщик сетевой услуги может удовлетворить запрос или сообщить, что запрошенная услуга недоступна.

Функции сетевого уровня обеспечивают использование различных конфигураций для поддержки сетевых соединений: от соединений, поддерживаемых двухпунктовыми сетевыми конфигурациями, до сетевых соединений, поддерживаемых сочетаниями подсетей с различными характеристиками.

Сетевой уровень выполняет следующие функции:

— маршрутизацию и ретрансляцию;
— организацию сетевых соединений;
— мультиплексирование сетевого соединения;
— сегментирование и объединение;
— обнаружение ошибок;
— восстановление при ошибках;
— упорядочение блоков данных;
— управление потоком данных;
— передачу срочных данных;
— сброс;
— выбор услуги;
— административное управление сетевым уровнем.

2.1.2.4. Уровень 4: Транспортный уровень

Транспортный уровень (Transport Layer) обеспечивает передачу данных без каких-либо изменений между сеансовыми логическими объектами и освобождает их от выполнения операций, обеспечивающих надёжную и экономически эффективную передачу данных.

Транспортный уровень оптимизирует использование доступных сетевых услуг, чтобы обеспечить пропускную способность, требуемую каждым сеансовым логическим объектом, при минимальных затратах. Эта оптимизация достигается путём внесения ограничений, обусловленных совместными требованиями со стороны всех одновременно работающих сеансовых логических объектов, а также общим качеством и объёмом сетевых услуг, предоставляемых транспортному уровню.

Все протоколы, определённые на транспортном уровне, имеют межоконечный характер. Под окончаниями понимают связанные транспортные логические объекты. Поскольку сетевые услуги обеспечивают сетевые соединения между транспортными логическими объектами по принципу «каждый с каждым», включая использование последовательно соединённых подсетей, то транспортный уровень освобождается от функций маршрутизации и ретрансляции.

На транспортном уровне имеются функции, обеспечивающие требуемое качество услуг на основе услуг, предоставляемых сетевым уровнем. Качество сетевых услуг зависит от того, как они реализуются.

Транспортный уровень однозначно идентифицирует каждый сеансовый логический объект с помощью транспортного адреса. Транспортные услуги предоставляют средства для установления, поддержания и разрыва транспортного соединения. Транспортное соединение обеспечивает дуплексную передачу между двумя транспортными адресами.

Для одной пары транспортных адресов может быть установлено несколько транспортных соединений. Сеансовые логические объекты используют идентификаторы оконечных пунктов транспортных соединений, обеспечиваемые транспортным уровнем для распознавания этих пунктов.

Качество услуг при предоставлении транспортного соединения зависит от класса обслуживания, запрашиваемого сеансовым логическим объектом при установлении транспортного соединения. Выбранное качество обслуживания поддерживается в течение существования транспортного соединения.

Транспортным уровнем предоставляются следующие виды услуг:
- установление транспортного соединения;
- передача данных;
- разрыв транспортного соединения.

На транспортном уровне могут быть реализованы следующие функции:
- преобразование транспортного адреса в сетевой;
- межоконечное мультиплексирование транспортных соединений в сетевые;
- установление и разрыв транспортных соединений;
- межоконечное упорядочение блоков данных по отдельным соединениям;
- межоконечное обнаружение ошибок и необходимый контроль за качеством услуг;
- межоконечное восстановление после ошибок;
- межоконечное сегментирование, объединение и сцепление;
- межоконечное управление потоком данных по отдельным соединениям;
- супервизорные функции;
- передача срочных транспортных сервисных блоков данных.

2.1.2.5. Уровень 5: Сеансовый уровень

Сеансовый уровень (Session Layer) реализует службу имён (отображение логических имён в сетевые адреса), устанавливает сеансы между службами и создаёт точки для контрольной синхронизации в случае потери связи.

Сеансовый уровень выполняет следующие функции:
- отображение сеансового соединения на транспортное соединение;
- управление потоком данных в сеансовом соединении;
- передачу срочных данных;
- восстановление сеансового соединения;
- административное управление сеансовым уровнем.

2.1.2.6. Уровень 6: Уровень представления

Уровень представления (Presentational Layer) устанавливает способы представления информации, которой обмениваются прикладные логические объекты или на которую они ссылаются в процессе этого обмена.

Уровень представления охватывает два взаимодополняющих аспекта способов представления информации:

— представление данных, подлежащих передаче между прикладными логическими объектами;
— представление структуры данных, которую прикладные логические объекты намереваются использовать в своём диалоге, наряду с представлениями совокупности действий, которые могут быть выполнены над этой структурой данных.

На этом уровне определяется общий синтаксис (способы представления данных), но не семантика, которая известна только прикладным логическим объектам.

Уровень представления обеспечивает способы представления информации, которые являются общими для взаимодействующих прикладных логических объектов. Таким образом, прикладные логические объекты освобождаются от функции представления информации, поскольку используется общий способ представления, и для них обеспечивается синтаксическая независимость. Такая независимость может быть реализована двумя путями.

1) На уровне представления обеспечиваются элементы поддержки синтаксиса, являющиеся общими для использующих их прикладных логических объектов.
2) Прикладные логические объекты могут использовать произвольный синтаксис, а уровень представления обеспечивает преобразование этих синтаксисов. Для обмена между прикладными логическими объектами применяется общий синтаксис. Такое преобразование выполняется внутри открытой системы. На другие открытые системы это не влияет и, следовательно, не оказывает влияние на стандартизацию протоколов уровня представления.

Уровень представления обеспечивает сеансовые услуги и добавляет к ним следующие возможности:

— преобразование синтаксиса;
— выбор синтаксиса.

Преобразование синтаксиса связано с преобразованием кодовых и символьных наборов, с модификацией расположения данных и с адаптацией действий над структурами данных. Выбор синтаксиса предоставляет средства первоначального выбора синтаксиса и последующего изменения сделанного выбора.

Прикладным логическим объектам предоставляются услуги сеансового уровня в виде услуг представления. На уровне представления выполняются следующие функции, с помощью которых реализуются услуги представления:

— запрос на установление сеанса;
— передача данных;
— соглашение по выбору и повторному выбору синтаксиса;
— преобразование синтаксиса, включая преобразование данных, форматирование и специальные функции преобразования;
— запрос на завершение сеанса.

2.1.2.7. Уровень 7: Прикладной уровень

Прикладной уровень (Application Layer) является наивысшим уровнем в эталонной модели OSI. Поэтому прикладной уровень не имеет интерфейса с более высоким уровнем. Он является единственным средством доступа прикладных процессов к функциональной среде OSI.

Прикладной уровень поддерживает локальные операционные системы, предоставляя им набор разнообразных протоколов, с помощью которых производится

доступ к сетевым ресурсам. Единицу информации на прикладном уровне называют *сообщением (Message)*.

Прикладные процессы обмениваются информацией с помощью прикладных логических объектов, прикладных протоколов и услуг уровня представления.

Прикладные услуги отличаются от услуг, предоставляемых другими уровнями, тем, что они не предоставляются какому-либо верхнему уровню и не связаны ни с каким пунктом доступа к услугам. Кроме передачи информации может предоставляться следующий набор услуг:

— идентификация партнёров, собирающихся инициировать связь;
— установление уровня авторизации для взаимодействия;
— авторизация партнёров, собирающихся инициировать взаимосвязь;
— определение параметров качества услуг, считающихся приемлемыми;
— идентификация ограничений на синтаксис данных;
— и другие.

На прикладном уровне выполняются все функции связи между открытыми системами, которые не выполняются нижележащими уровнями. В их число включаются функции, выполняемые программными средствами, и функции, выполняемые людьми.

2.2. Иерархия протоколов в различных стеках

2.2.1. Стек ISO/OSI

В данном случае эталонная модель первична, а стек протоколов вторичен. Это привело к некоторой тяжеловесности протоколов данного стека (рис. 2.2). Интересно, что большое количество протоколов стека разработано под влиянием IBM.

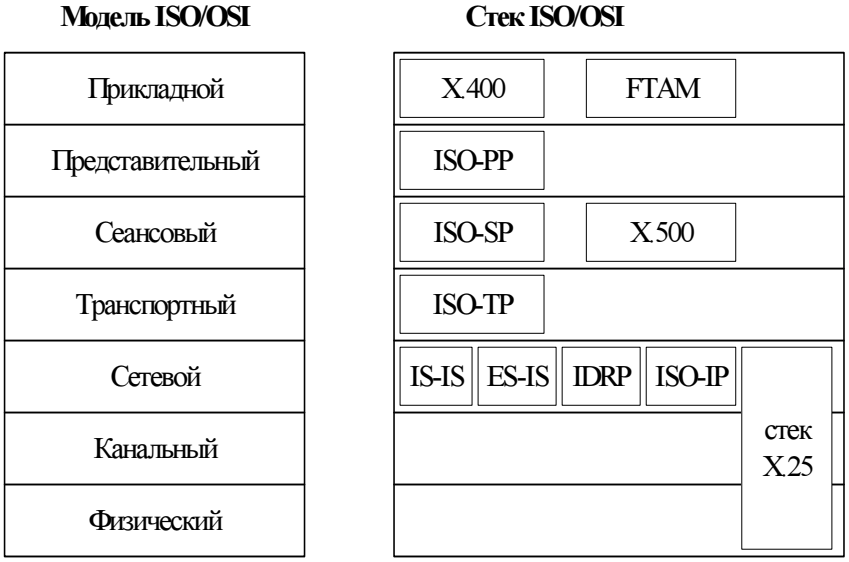


Рис. 2.2. Некоторые протоколы стека ISO/OSI

Из-за ограничений определения *физического уровня* в эталонной модели протоколы физического уровня в данном стеке практически отсутствуют (за исклю-

чением семейства протоколов X.25, которое, впрочем, по генезису выбивается из общего построения стека протоколов ISO/OSI).

К *канальному уровню* можно отнести протокол LLC, который хотя и разработан в рамках IEEE 802.2, но служит для сопряжения стека протоколов ISO/OSI с канальным уровнем других стеков.

Основным протоколом *сетевого уровня* является *протокол межсетевого взаимодействия ISO (ISO Internetworking Protocol, ISO-IP)*, описанный в RFC 1575 [1] и документах ISO S 8473, IS 8348. Другое его название — *услуга организации сетевого взаимодействия без установления соединения (Connectionless Network Service, CLNS)*.

Функцию маршрутизации обеспечивают протоколы *IS-IS (Intermediate System to Intermediate System)* (ISO 10589), *ES-IS (End System to Intermediate System)* (ISO 9542) и *CLNS* (ISO 8473), а также *внутридоменный протокол маршрутизации (Inter Domain Routing Protocol, IDRP)* (ISO 7498).

На *транспортном уровне* располагается *транспортный протокол ISO (ISO Transport Protocol, ISO-TP)* (ISO 8073).

Основным протоколом *сеансового уровня* является протокол *ISO-SP (OSI Session Layer Protocol)* (соответствует спецификации ISO/IEC 8327-1 09-1996 ITU-T X.225.) На этом же уровне находится *протокол доступа к каталогам X.500* (прародитель протокола LDAP стека TCP/IP). Кроме того, следует отметить *протокол ISO NetBIOS* (соответствует протоколу NetBIOS одноимённого стека протоколов).

На уровне представления находится *протокол представления (Presentation Protocol, PP)* (ISO IS 8823).

На *прикладном уровне* присутствует набор протоколов, достаточный для основных пользовательских приложений. Здесь же следует упомянуть почтовые протоколы X.400, базирующиеся на рекомендациях CCITT с X.400 по X.430. Стандарт X.400 описывает функционирование *агентов передачи почты (Message Transfer Agents, MTA)*.

Доступ к файлам описывается *протоколом управления доступом и передачей файлов (File Transfer Access and Management, FTAM)* (аналог FTP в стеке TCP/IP). Кроме того, на этом уровне находится сетевой *протокол разделения файлов (Server Message Block, SMB)*.

2.2.2. Стек TCP/IP

Эталонная модель TCP/IP документирует дизайн семейства протоколов TCP/IP и состоит из четырёх уровней (рис. 2.3, 2.4).

Основой модели служит *межсетевой уровень*. Его задачей является доставка пакетов в пункт назначения. Передача осуществляется без установления соединения. Здесь же осуществляется выбор маршрута пакета. Пакеты могут двигаться к пункту назначения разными маршрутами, поэтому и прибывать они могут не в том порядке, в котором были отправлены.

На межсетевом уровне определён протокол *IP (Internet Protocol)*, задающий в том числе и схему адресации. Кроме того, здесь же определены протоколы маршрутизации *RIP (Routing Information Protocol)*, *OSPF (Open Shortest Path First)*, *BGP (Border Gateway Protocol)*.

Таким образом, межсетевой уровень модели TCP/IP близок сетевому уровню эталонной модели OSI.

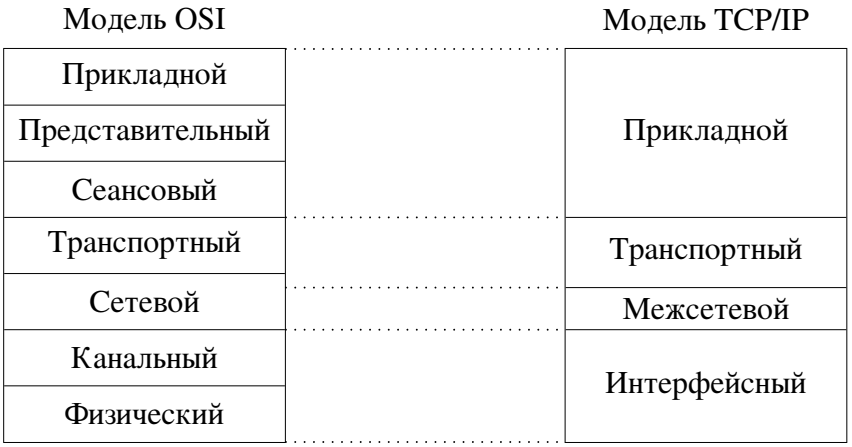


Рис. 2.3. Соответствие эталонных моделей OSI и TCP/IP

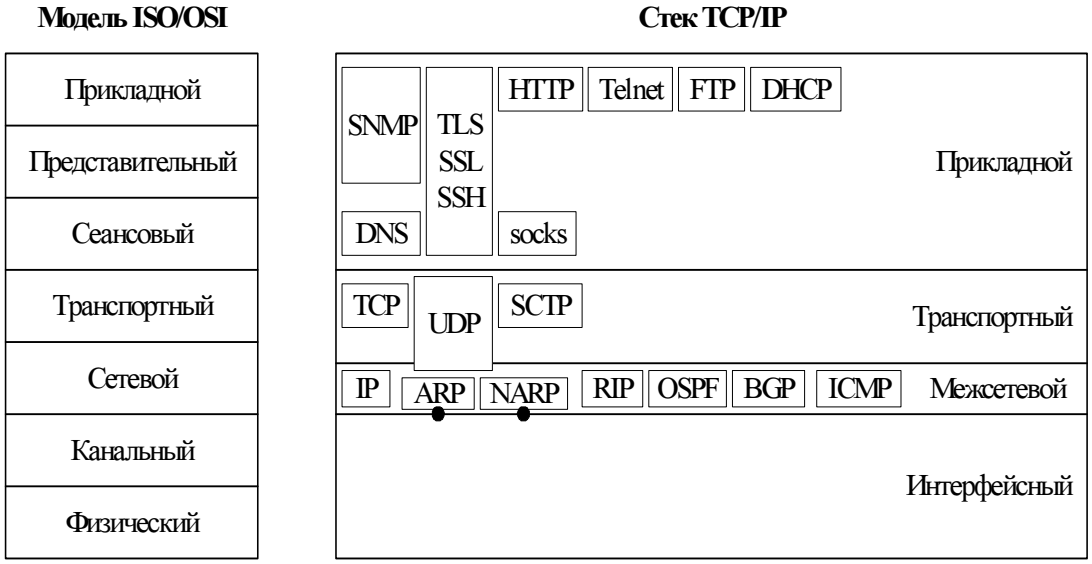


Рис. 2.4. Некоторые протоколы стека TCP/IP

На *транспортном уровне* модели TCP/IP решается задача поддержания связи между отправителем и получателем. Этот уровень в основном соответствует транспортному уровню эталонной модели OSI. На нём определены протоколы *TCP (Transmission Control Protocol)*, *UDP (User Datagram Protocol)*, *DCCP (Datagram Congestion Control Protocol)*, *SCTP (Stream Control Transmission Protocol)*.

Прикладной уровень объединяет все службы, представляемые системой пользовательским приложениям. В модели TCP/IP не выделяются отдельно *сеансовый* и *представительный* уровни. Отдельные их функции выполняются различными протоколами прикладного уровня. На этом уровне определены, например, почтовые протоколы *SMTP (Simple Network Management Protocol)*, *IMAP4 (Internet Message Access Protocol rev 4)*, *POP3 (Post Office Protocol version 3)*, протокол передачи гипертекста *HTTP (Hypertext Transfer Protocol)*, протокол передачи файлов *FTP (File Transfer Protocol)*, протокол эмуляции терминала *Telnet* и др.

Интерфейсный уровень отвечает за взаимодействие между компьютером и физическим сетевым оборудованием. Он приблизительно соответствует канальному и физическому уровням модели OSI. Интерфейсный уровень по-настоящему не описан в документации по архитектуре TCP/IP, в которой сказано только, что он обеспечивает доступ к сетевой аппаратуре системно-зависимым способом.

2.2.3. Стек IEEE 802

Семейство протоколов IEEE 802 базируется на фирменных стандартах построения локальных сетей Arcnet, Ethernet, Token Ring.

Протоколы IEEE 802 охватывают только два нижних уровня семиуровневой эталонной модели OSI, а именно физический и канальный (рис. 2.5). Это связано с тем, что именно эти уровни в наибольшей степени отражают специфику локальных сетей.

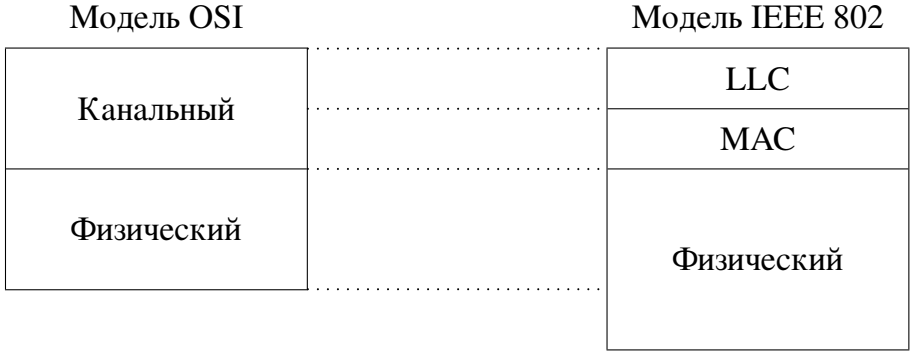


Рис. 2.5. Соответствие эталонных моделей OSI и IEEE 802

На *физическом уровне* модели IEEE 802 специфицируются также и различные типы носителей, то есть среда передачи, что не входит в определение физического уровня эталонной модели OSI. Поэтому физический уровень модели IEEE 802 изображён охватывающим область, лежащую ниже физического уровня модели OSI.

В спецификации IEEE *канальный уровень (Data Link Control, DLC)* разделяется на уровень *управления логическим каналом (Logical Linc Control, LLC)* и уровень *управления доступом к носителю (Media Access Control, MAC)*. По сути, уровень MAC эквивалентен всему уровню DLC в предыдущих спецификациях. Добавление уровня LLC является результатом давления IBM, разрабатывавшей стандарт Token Ring одновременно со спецификацией IEEE 802.5. Поэтому уровень LLC — это отражение операций *высокоуровнего протокола управления каналом передачи данных (High-Level Data Link Control, HDLC)* в *системной сетевой архитектуре (Systems Network Architecture, SNA)*.

2.2.4. Стек IPX/SPX

Стек протоколов IPX/SPX (или стек Novell NetWare) разработан в начале 1980-х гг. фирмой Novell для сетевой операционной системы NetWare.

Стек включает в себя следующие протоколы (рис. 2.6): *протокол межсетевого обмена (Interwork Packet Exchange, IPX)*, *протокол маршрутизации (Rout-*

ing Information Protocol, RIPX)[1], *протокол упорядоченного обмена пакетами (Sequenced Packet Exchange, SPX)*, *протокол анонсирования сервиса (Service Advertising Protocol, SAP)*, *протокол ядра NetWare (Netware Core Protocol, NCP)* и др.

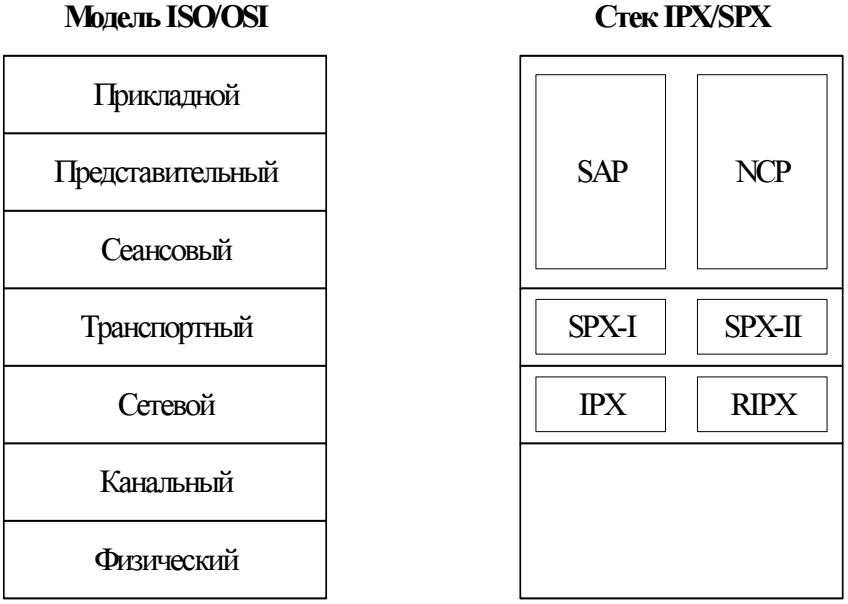


Рис. 2.6. Некоторые протоколы стека IPX/SPX

2.2.5. Стек NetBIOS/SMB

Стек NetBIOS/SMB разработан в 1984 г. совместно IBM и Microsoft для сетей IBM PC Network и IBM Token Ring.

Протокол *NetBIOS (Network Basic Input/Output System)* был разработан как аналог системы BIOS персонального компьютера. Он реализует большинство услуг и функций сетевого, транспортного и сеансового уровней модели ISO/OSI (так, протокол NetBIOS не поддерживает маршрутизацию пакетов, что является одной из основных функций сетевого уровня). Однако впоследствии за протоколом NetBIOS остался только сеансовый уровень, поскольку на более низких уровнях стали использовать стандартные протоколы (например, TCP/IP или IPX/SPX).

Следует отметить, что существует три реализации протокола NetBIOS:

— *NetBEUI (NetBIOS Extended User Interface)* — NetBIOS поверх LLC;
— *NBT (NetBIOS over TCP/IP)* — NetBIOS поверх IP;
— *NetBIOS* — NetBIOS поверх IPX.

Протокол *SMB (Server Message Block)* реализует услуги и функции прикладного уровня и уровня представления модели ISO/OSI. Протокол регламентирует взаимодействие рабочей станции с сервером. В его функции входит создание и разрыв логического соединения между рабочей станцией и сетевыми ресурсами файлового сервера, управление доступом к файлам на файловом сервере, управление очередью печати на сервере печати.

[1]Следует различать протокол RIP в стеке TCP/IP и протокол RIPX в стеке IPX/SPX.

2.2.6. Стек H.323

Стандарт H.323 входит в серию рекомендаций H.32x ITU-T, разработанных для регламентации проведения аудио- и видеоконференций по телекоммуникационным сетям:

— H.320 регламентирует организацию мультимедийной связи по сетям ISDN;
— H.321 регламентирует организацию мультимедийной связи по сетям ATM;
— H.322 регламентирует организацию мультимедийной связи по сетям с коммутацией пакетов с гарантированной пропускной способностью;
— H.323 регламентирует организацию мультимедийной связи по сетям с коммутацией пакетов с негарантированной пропускной способностью;
— H.324 регламентирует организацию мультимедийной связи по телефонным сетям общего пользования;
— H.324/C регламентирует организацию мультимедийной связи по сетям мобильной связи.

По сути, H.323 является набором управляющих протоколов (рис. 2.7), строго регламентирующих использование программ (кодеков) и протоколов других стеков для организации мультимедийной связи по сетям с коммутацией пакетов.

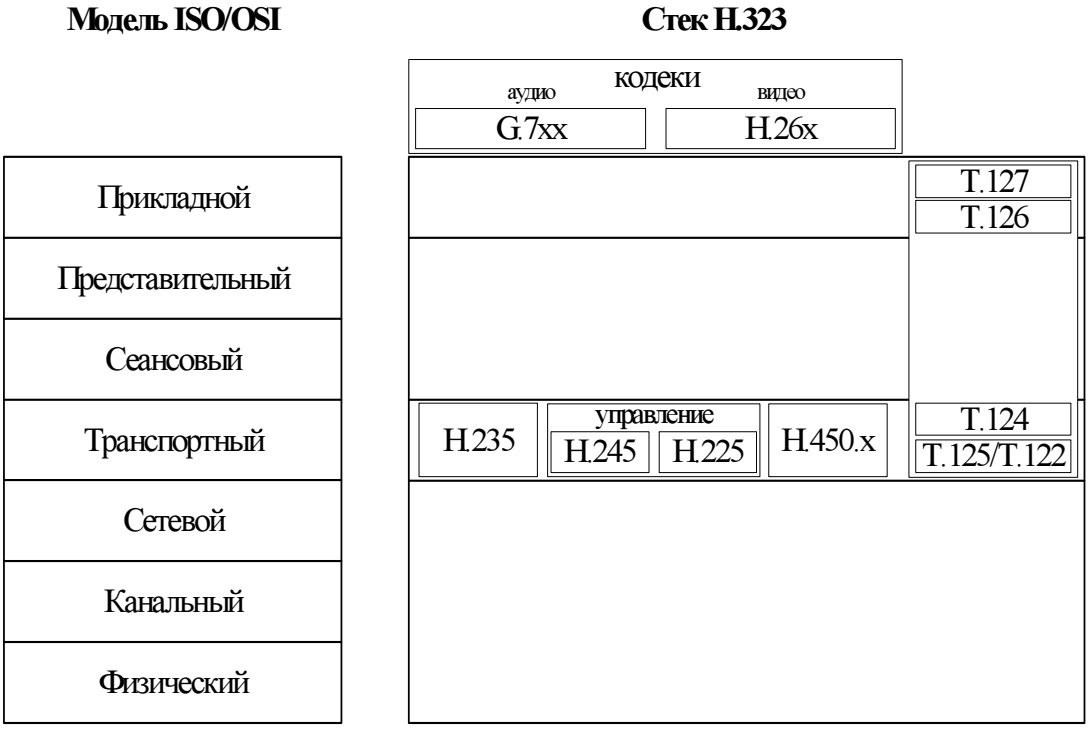


Рис. 2.7. Некоторые протоколы стека H.323

За управление соединением и сигнализацией отвечают следующие протоколы:

— H.225.0 — протокол сигнализации и пакетирования мультимедийного потока;
— H.225.0/RAS — протокол, определяющий процедуры регистрации, доступа и состояния;
— H.245 — протокол управления для мультимедиа.

За безопасность и шифрование отвечает протокол H.235.

Протоколы H.450.x определяют различные дополнительные услуги:

- H.450.1 — определяет функции для управления дополнительными услугами;
- H.450.2 — осуществляет перевод соединения третьему абоненту;
- H.450.3 — осуществляет переадресацию вызова;
- H.450.4 — осуществляет удержание вызова;
- H.450.5 — осуществляет прикрепление вызова (park) и ответ на вызов (pick up);
- H.450.6 — осуществляет уведомление о вызове в режиме разговора;
- H.450.7 — осуществляет индексацию ожидающего сообщения;
- H.450.8 — осуществляет идентификацию имён;
- H.450.9 — осуществляет завершение соединения.

За организацию конференц-связи для передачи данных отвечает стек T.120, включающий в себя протоколы T.123, T.124, T.125.

Для обработки аудиосигнала применяются кодеки серии G.7xx: G.711, G.722, G.723.1, G.728, G.729.

Для обработки видеосигнала используются кодеки H.261, H.263, H264.

2.2.7. Стек SS7

Система сигнализации № 7 (SS7 — Signaling System 7, или ОКС7 — система общеканальной сигнализации № 7) разработана и стандартизована ITU-T в 1981 г. и представляет собой набор протоколов сигнализации, предназначенных для обмена информацией управления вызовами между коммутационными станциями и специализированными узлами сетей связи для поддержки как голосовых, так и неголосовых служб [2, 3]. SS7 образует собственную сеть, работающую параллельно цифровой сети связи.

Стек SS7 имеет четыре уровня, соответствующие физическому, канальному, сетевому и прикладному уровням модели ISO/OSI (рис. 2.8).

Подсистема передачи сообщений (Message Transfer Part, MTP) состоит из трёх уровней — *MTP1*, *MTP2*, *MTP3*, образующих общую транспортную подсистему, обеспечивающую корректную передачу информации между узлами сети сигнализации.

Уровень MTP1 соответствует физическому уровню модели ISO/OSI. На нём определены физические, электрические и функциональные характеристики звена данных сигнализации и средства доступа к нему.

Уровень MTP2 соответствует канальному уровню модели ISO/OSI. На нём определены функции и процедуры, относящиеся к передаче сигнальных сообщений по отдельному звену сигнализации.

На уровне MTP3 определены процедуры и функции сети сигнализации по маршрутизации сообщений, есть возможность восстановления способности передачи сигнальных сообщений после сбоев в сети, но лишь частично поддерживается адресация. Поэтому данный уровень лишь частично можно соотнести с сетевым уровнем модели ISO/OSI. Соответствие уровней становится полным, если рассматривать данный уровень совместно с *подсистемой управления соединениями сигнализации (Signaling Connection Control Part, SCCP)*.

Подсистемы MTP и SCCP в совокупности образуют *подсистему сетевых услуг (Network Service Part, NSP)*.

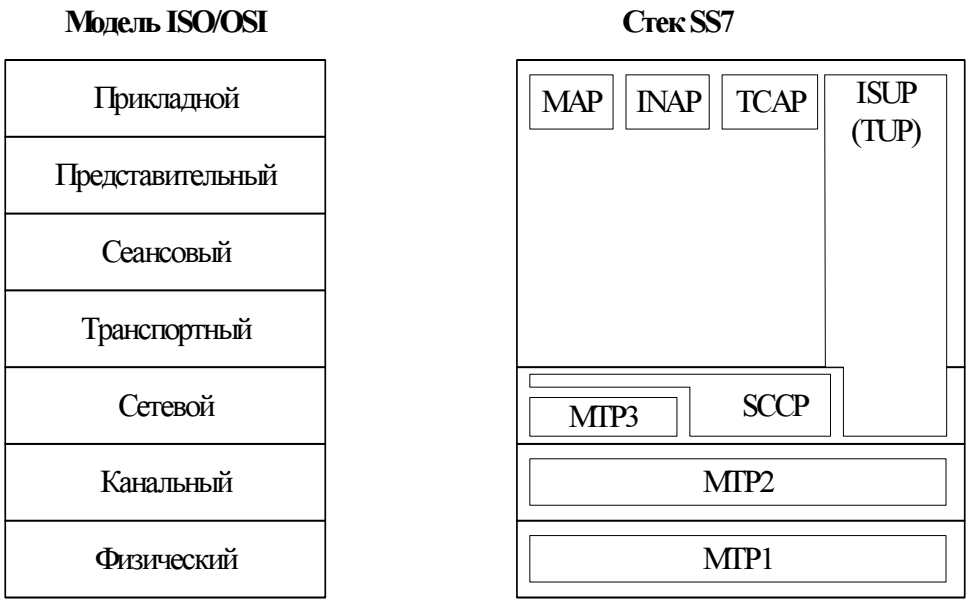


Рис. 2.8. Некоторые протоколы стека SS7

Протокол *ISUP (ISDN User Part)* определяет сигнальные функции для установления соединений с возможностью предоставления услуг *цифровой сети с интеграцией служб (Integrated Service Digital Network, ISDN)*. Ранее функции по управлению вызовами выполняла подсистема *TUP (Telephone User Part)*, впоследствии полностью вошедшая в ISUP. По отношению к модели ISO/OSI ISUP занимает сетевой и прикладной уровни.

На прикладном уровне модели ISO/OSI располагаются прикладная *подсистема обеспечения транзакций (Transaction Capabilities Applications Part, TCAP)*, *подсистема пользовательской мобильной связи (Mobile Application Part, MAP)* и *протокол интеллектуальной сети (Intelligent Network Application Protocol, INAP)*.

Глава 3. Физический уровень

Напомним, что на физическом уровне модели ISO/OSI обеспечивается передача битовых потоков без каких-либо изменений между логическими объектами уровня звена данных по физическим соединениям, специфицируются соединители, но не сама среда, определяются базовые механизмы кодирования и декодирования двоичных данных в физическом носителе.

Таким образом, среда, согласно эталонной модели, рассматривается как нечто, лежащее ниже физического уровня, а битовый поток в носителе должен быть независим от типа среды.

3.1. Среда передачи

Среда передачи данных:

- кабельная:
  - коаксиальный кабель:
    - 10Base-5 — толстый коаксиальный кабель,
    - 10Base-2 — тонкий коаксиальный кабель;
  - витая пара:
    - неэкранированная:
      - UTP (Unshielded Twisted Pair) — витая пара без индивидуального экранирования;
    - экранированная[1]:
      - FTP (Foiled Twisted Pair) — фольгированная витая пара (общий фольгированный экран, без индивидуального экранирования витой пары),
      - STP (Shielded twisted Pair) — защищённая витая пара (каждая пара имеет экран),
      - ScTP (Screened Twisted Pair) — экранированная витая пара;
- оптоволокно:
  - одномодовое,
  - многомодовое;
- эфир:
  - ДВ-, СВ-, КВ-, УКВ-связь без применения ретрансляторов,
  - радиорелейная связь,
  - спутниковая связь,
  - сотовая связь.

[1] Экран может быть выполнен либо из токопроводящей фольги (блокирует высокочастотное электромагнитное излучение), либо из переплетённой медной проволоки (хорошо защищает от низкочастотных наводок).

3.1.1. Основные характеристики среды передачи

Факторы, влияющие на передачу сигнала:

— помехи или шумы — любой нежелательный сигнал, который смешивается с сигналом, предназначенным для передачи или приёма, и искажает его;
— скорость передачи данных — скорость в битах в секунду (бит/с), с которой могут передаваться данные;
— ширина полосы — ширина полосы передаваемого сигнала, ограничиваемая передатчиком и природой передающей среды (выражается в периодах в секунду, или герцах (Гц));
— пропускная способность канала — максимально возможная при определённых условиях скорость, при которой информация может передаваться по конкретному тракту связи или каналу;
— уровень ошибок — частота появления ошибок (ошибкой считается приём 1 при переданном 0, и наоборот).

3.1.2. Коаксиальный кабель

В современных сетях коаксиальный кабель не используется из-за дороговизны и трудностей эксплуатации. Тем не менее в данном разделе приведено краткое представление о нём.

Коаксиальный кабель — вид электрического кабеля, состоящего из двух цилиндрических проводников, соосно вставленных один в другой. В центре — медный проводник, покрытый пластиковым изолирующим материалом, поверх — медная сетка или алюминиевая фольга, покрытая внешней оболочкой.

Особенности сетей, использующих толстый коаксиальный кабель и протокол передачи данных 10Base5:

— стандарт 10Base5 поддерживает до 100 узлов на сегмент (расстояние между узлами кратно 2,5 м);
— максимальная длина сегмента — не более 500 м;
— в сети не может быть более 5 сегментов, 4 повторителей, и только 3 сегмента могут иметь подключённые устройства;
— ограничение скорости в 10 Мбит;
— рассоединение шины в любом месте полностью нарушает работоспособность сети;
— низкая устойчивость к статическому напряжению и грозовым наводкам.

Особенности сетей, использующих тонкий коаксиальный кабель и протокол передачи данных 10Base2:

— к одному сегменту может быть подключено не более 30 устройств, минимальное расстояние между которыми составляет 0,5 м;
— максимальная длина сегмента — не более 185 м;
— к локальной сети может быть подключено максимум 90 компьютеров;
— для подключения сетевых адаптеров к кабелю используются специальные Т-коннекторы (T-Connector);
— ограничение скорости в 10 Мбит;
— рассоединение шины в любом месте полностью нарушает работоспособность сети;
— низкая устойчивость к статическому напряжению и грозовым наводкам.

3.1.3. Витая пара

Витая пара — вид кабеля связи, представляющий собой одну или несколько пар изолированных проводников, скрученных между собой с целью уменьшения взаимных наводок при передаче сигнала. Для магистральных линий часто используют кабели с 10, 25, 50, 100 и более, парами в одной оболочке.

Как было отмечено выше, витая пара может быть экранированной и неэкранированной. Экран может быть или плетёным из медной проволоки, что хорошо защищает от низкочастотных наводок, или из токопроводящей фольги (плёнки), которая блокирует высокочастотное электромагнитное излучение.

Категория (Category) витой пары определяет частотный диапазон, в котором её применение эффективно (табл. 3.1). Категории определяются стандартом EIA/TIA 568A. В последней графе приводится классификация линий связи, обеспечиваемых этими кабелями, по стандарту ISO 11801 и EN 50173.

Таблица 3.1

Классификация витой пары по категориям

Категория	Полоса частот, МГц	Число пар	Тип сети	Скорость
1	0, 1	1	Аналоговая телефонная сеть	-
2	1	2	ISDN, Token Ring, ARCNet	до 4 Мбит/с
3	16	2	Ethernet, Token Ring	10 Мбит/с (10Base-T) или 100 Мбит/с (100Base-T4)
4	20	4	Token Ring	16 Мбит/с по одной паре (10BASE-T, 100BASE-T4)
5	100	4	Fast Ethernet	до 100 Мбит/с (100Base-TX — используются 2 пары)
5e	125	4	Gigabit Ethernet	до 1 Гбит/с (1000Base-TX — используются 4 пары)
6	200 (250)	4	Fast Ethernet, Gigabit Ethernet	до 10 Гбит/с
7	600–700	4	-	до 100 Гбит/с

3.1.4. Оптоволокно

Оптический кабель состоит из некоторого количества оптических волокон, окружённых общей защитной оболочкой. Оптическое волокно состоит из серд-

цевины, оптической оболочки, защитного покрытия, буферного покрытия.

Источником распространяемого по оптическим кабелям света является светодиод (или полупроводниковый лазер), а кодирование информации осуществляется двухуровневым изменением интенсивности света (0–1). На другом конце кабеля принимающий детектор преобразует световые сигналы в электрические.

Различают *одномодовое* и *многомодовое* оптоволокно. Многомодовое и одномодовое оптоволокно отличаются способом распространения оптического излучения в волокне.

Многомодовое волокно имеет диаметр сердечника почти на два порядка больше, чем длина световой волны (обычно 50 или 62,5 мкм), т.е. свет может распространяться в волокне по нескольким независимым путям (модам), причём разные моды имеют разную длину.

Таблица 3.2

Сравнение одномодовых и многомодовых технологий

Параметры	Одномодовые	Многомодовые
Используемые длины волн	$1,3$ и $1,5$ мкм	$0,85$ мкм, реже $1,3$ мкм
Затухание, дБ/км	$0,4$–$0,5$	$1,0$–$3,0$
Тип передатчика	лазер, светодиод	светодиод
Толщина сердечника	8 мкм	50 или $62,5$ мкм
Дальность передачи Fast Ethernet	около 20 км	до 2 км
Дальность передачи специально разработанных устройств Fast Ethernet	более 100 км	до 5 км
Возможная скорость передачи	10 Гбит и более	до 1 Гбит на ограниченной длине
Область применения	телекоммуникации	локальные сети

Оптоволокно является традиционной физической средой передачи данных по магистральным сетям. В этом случае способы его применения классифицируют по названию точки сопряжения с потребителем и объединяют названием типа FTTx — оптоволокно до точки «x», например: оптика до административного здания (Fiber To The Building, FTTB), до распределительного шкафа (Fiber To The Curb, FTTC), до жилого дома (Fiber To The Home, FTTH), до некоторого выносного модуля (Fiber To The Remote, FTTR).

3.1.5. Структурированная кабельная система

Кабельная система является основой функционирования любой компьютерной сети. Надёжность и расширяемость всей сети напрямую зависят от качества

кабельной системы, на которой она работает. Для решения проблемы эффективности кабельных коммуникаций служат структурированные кабельные системы.

Структурированная кабельная система (СКС) — это иерархическая кабельная система, состоящая из нескольких стандартизованных подсистем. Фактически СКС представляет собой набор кабелей, соединительных элементов, кроссировочных панелей, розеток, монтажных шкафов и коробов. СКС включает в себя не только сетевые кабели, но и телефонные, а также кабели систем видеонаблюдения, сигнализации и др.

Важным принципом построения СКС является избыточность. Число розеток и другого оборудования, устанавливаемого при прокладке кабельной системы, часто намного превышает необходимое в данный момент. Такой подход позволяет в дальнейшем увеличивать нагрузку и производить переконфигурацию сети.

СКС имеет чёткую иерархическую структуру. В общем случае в СКС выделяют следующие основные подсистемы:

- *горизонтальную* — используется для подключения рабочих мест пользователей и оборудования в пределах этажа здания;
- *вертикальную* (или магистраль здания) — служит для подключения горизонтальных подсистем друг к другу (соединяет распределительные панели этажей с распределительной панелью здания);
- *внешнюю* (или магистраль комплекса) — служит для подключения вертикальных подсистем друг к другу и представляет собой кабельную магистраль (как правило, оптоволоконную), объединяющую оборудование нескольких зданий, находящихся на расстоянии до нескольких километров друг от друга.

Для проводки кабеля в горизонтальной и вертикальной подсистемах принята топология *звезда*. В горизонтальной подсистеме, где наибольшее число ответвлений кабеля, предпочтительно использовать кабели из неэкранированной витой пары (UTP) категории 5. Для обеспечения надёжности такой системы следует применять розетки, разъёмы и патч-панели, также удовлетворяющие требованиям категории 5. В вертикальных подсистемах, согласно общим рекомендациям, применяются оптоволоконные кабели или же экранированная витая пара (STP). Во внешней подсистеме между зданиями целесообразно использовать оптоволокно, так как оно обеспечивает наибольшую допустимую длину сегмента и пропускную способность. Коаксиальный кабель в СКС не применяется.

Для соединения телекоммуникационного оборудования используется спецификация физического интерфейса *RJ*[1] *(Registered Jack)* (FCC, Part 68, Subpart F, Section 68.502 [4]). Стандартные варианты этого разъёма называются RJ-11, RJ-14, RJ-25, RJ-45 и так далее. Разъёмы RJ (рис. 3.2) принадлежат к семейству модульных разъёмов, за исключением RJ-21.

Термин *RJ-45* ошибочно употребляется для обозначения разъёма 8P8C, используемого в компьютерных сетях. На самом деле настоящий RJ-45 физически несовместим с 8P8C (8 контактов, 8 проводников), так как использует схему 8P2C (8 контактов, 2 проводника) с ключом. Ошибочное употребление термина RJ-45 вызвано, вероятно, тем, что настоящий RJ-45 не получил широкого применения, а также их внешним сходством.

[1] С этими стандартами связана большая путаница. Шестиместный разъём, часто применяемый в телефонии, может быть использован как RJ-11, RJ-14 или даже RJ-25, которые по сути являются названиями стандартов, использующих этот физический разъём. RJ-11 предполагает двужильное соединение, в то время как RJ-14 — четырёхжильное, а RJ-25 использует все шесть жил.

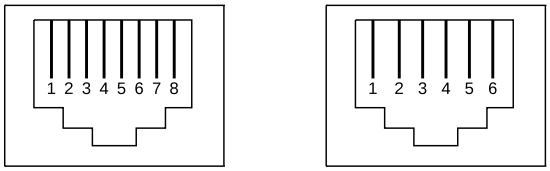


Рис. 3.1. Модульные розетки

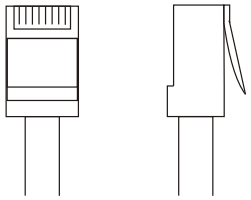


Рис. 3.2. Общий вид разъёма RJ-45

Для соединения оборудования в компьютерных и телефонных сетях чаще всего применяются восьмиконтактные модульные разъёмы RJ-45/8P8C (компьютерные), шестиконтактные RJ-12/6P6C (телефонные) и четырехконтактные RJ-11/6P4C (телефонные).

Четырёхконтактный модульный разъём RJ-11 используется в телефонии для соединения телефонных аппаратов с телефонными трубками. Шестиконтактный модульный разъём RJ-12 — в основном для соединения телефонных аппаратов с розеткой. Разъёмы RJ-11 и RJ-12 применяются с плоским 1–3-парным телефонным кабелем. При подключении шестиконтактного разъёма RJ-12 к аналоговому телефонному аппарату используются только два центральных контакта.

Использование контактов модульных соединителей, а также цветовая маркировка проводов стандартизованы. Каждая пара представляется двумя проводами, обозначаемыми Tip и Ring (условно — прямой и обратный провода), для которых определены цвет изоляции и номер контакта разъёма. Для обозначения пар кабеля используется цветовая маркировка (табл. 3.3).

Таблица 3.3

Цветовая маркировка витой пары

№ пары	Цвет: основной / полоски
1	синий / бело-синий
2	оранжевый / бело-оранжевый
3	зелёный / бело-зелёный
4	коричневый / бело-коричневый

Для разводки четырёхпарного кабеля UTP в разъёмах RJ-45 стандартом EIA/TIA-568 приняты две основные схемы распределения пар проводов по контактам: EIA/TIA-T568A и EIA/TIA-T568B (рис. 3.3, табл. 3.4).

Таблица 3.4

Разводка контактов по схемам EIA/TIA-T568A и EIA/TIA-T568B

EIA/TIA-T568A			EIA/TIA-T568B		
№	Цвет	Пара	№	Цвет	Пара
1	бело-зелёный	3 (Tip)	1	бело-оранжевый	2 (Tip)
2	зелёный	3 (Ring)	2	оранжевый	2 (Ring)
3	бело-оранжевый	2 (Tip)	3	бело-зелёный	3 (Tip)
4	синий	1 (Ring)	4	синий	1 (Ring)
5	бело-синий	1 (Tip)	5	бело-синий	1 (Tip)
6	оранжевый	2 (Ring)	6	зелёный	3 (Ring)
7	бело-коричневый	4 (Tip)	7	бело-коричневый	4 (Tip)
8	коричневый	4 (Ring)	8	коричневый	4 (Ring)

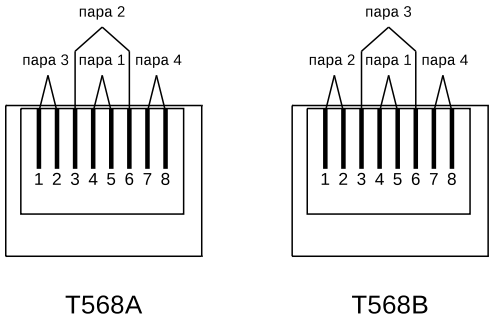


Рис. 3.3. Разводка контактов по схемам EIA/TIA-T568A и EIA/TIA-T568B

Для соединения двух компьютеров (устройств) витой парой напрямую, без применения каких-либо дополнительных(промежуточных) устройств, служит «перекрёстный» (кроссовый) кабель. Концы такого кабеля обжимаются по разным стандартам (один EIA/TIA-568A, другой EIA/TIA-568B) (рис. 3.4).

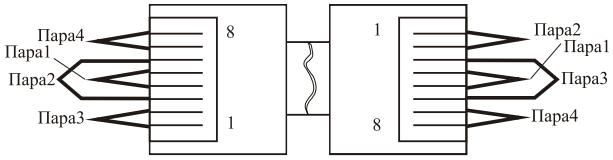


Рис. 3.4. Разводка кроссового кабеля

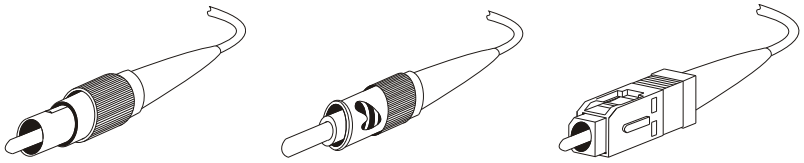
Для подключения оптических кабелей применяются разъёмы трёх основных типов — FC, ST и SC (см. рис. 3.5а–3.5в). Разъём FC применяется в основном в одномодовых системах.

Разъём SC используется как в многомодовых, так и в одномодовых системах. Этот оптический разъём рекомендуется как основной тип разъёма для применения в СКС. Может быть одинарным или двойным (дуплексным). Квадратный разъём снабжён защёлкой с фиксатором.

Разъём ST рекомендуется в первую очередь для многомодовых систем. Оптический разъём фиксируется в розетке подпружиненным байонетным элементом.

Преимущества структурированной кабельной системы:

— длительный срок эксплуатации без модернизации (10–15 лет);
— надёжность;
— возможность наращивания мощности и лёгкость расширения сети без изменения существующей сети;
— универсальная среда передачи (единая кабельная система для передачи данных, голоса и видеосигнала).



(а) Оптический разъём FC

(б) Оптический разъём ST

(в) Оптический разъём SC

Рис. 3.5. Оптические разъёмы

Таким образом, СКС является современным сетевым решением, обеспечивающим здание надёжными и многофункциональными коммуникациями на довольно длительный срок.

3.2. Активное сетевое оборудование

Активные устройства осуществляют формирование, преобразование, коммутацию, а также приём сигнала с использованием внешнего (не передающегося в составе сигнала) источника энергии. Активные устройства можно, с некоторой долей условности, разделить на рабочие станции, повторители, концентраторы, коммутаторы, мосты и маршрутизаторы.

Повторитель (Repeator) представляет собой устройство для физического соединения двух или более сегментов кабеля локальной сети с целью увеличения общей длины сети.

В сетях на витой паре и оптоволокне повторитель является самым дешёвым вариантом связующего устройства и чаще называется концентратором.

Концентратор (Hub) представляет собой многопортовый повторитель с автосегментацией. Каждый порт имеет собственный трансивер — приёмник, передатчик и детектор коллизий. Получив сигнал от одной из подключённых к нему станций, концентратор транслирует его на все свои активные порты. Если на каком-либо из портов обнаружена неисправность, то этот порт автоматически отключается (сегментируется), а после её устранения снова становится активным.

Повторитель работает на уровне физических сигналов — закодированных битовых цепочек, анализ кадров не выполняется.

Сетевой коммутатор (Switch) — устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного сегмента. Коммутатор хранит в памяти таблицу MAC-адресов, в которой указывается соответ-

ствие MAC-адреса узла порту коммутатора. Коммутатор передаёт данные непосредственно получателю.

Мост (Bridge) делит разделяемую среду передачи сети на логические сегменты, передавая информацию из одного сегмента в другой только в том случае, если адрес узла назначения принадлежит другой подсети. Таким образом трафик одной подсети изолируется от трафика другой подсети, что увеличивает пропускную способность сети в целом и уменьшает возможность несанкционированного доступа в подсеть.

Маршрутизатор (Router) осуществляет связь разных типов сетей и обеспечивает доступ к глобальной сети, управляет трафиком на основе протокола сетевого уровня. Подобно повторителям, маршрутизаторы восстанавливают уровень и форму передаваемого сигнала. Так же, как и мосты, они не передают адресату коллизии или повреждённые кадры, и из-за буферизации имеют задержку при передаче. Но в отличие от повторителей, мостов и коммутаторов, маршрутизаторы переформировывают передаваемые кадры Ethernet, а также могут поддерживать такие нетиповые функции, как подсчёт трафика, авторизация пользователей, ведение статистики и т.п.

Шлюз (Gateway) соединяет отдельные сегменты сети с разными типами системного и прикладного программного обеспечения.

Коммутаторы 3-го уровня (*маршрутизирующие коммутаторы* или *коммутирующие маршрутизаторы*) строятся на распределённой архитектуре — каждый порт имеет собственный специализированный процессор, отвечающий за анализ кадров и пакетов для определения их точки назначения, и общий управляющий процессор. Кадры, приходящие в порт и адресуемые (MAC-адресами) узлам той же подсети, но подключённым к другим портам, коммутируются (IP-заголовок не используется и не модифицируется). Кадры, приходящие на MAC-адрес порта, маршрутизируются — порт назначения определяется по IP-адресу назначения). Отличие от комбинации отдельного коммутатора с обычным маршрутизатором заключается в масштабировании пропускной способности каждой подсети: чем больше портов в неё входит, тем выше пропускная способность. Кроме того, и при коммутации может использоваться информация 3-го уровня (например, для фильтрации или приоритизации). Коммутаторы 3-го уровня в основном предназначены для организации связи подсетей в локальных сетях, и интерфейсов глобальных сетей они могут и не иметь.

3.3. Модуляция сигналов

Необходимость в модуляции аналоговой информации возникает при передаче низкочастотного (например, голосового) аналогового сигнала через канал, находящийся в высокочастотной области спектра. Для решения этой проблемы амплитуду высокочастотного несущего сигнала изменяют (модулируют) в соответствии с изменением низкочастотного сигнала.

Основные технологии модуляции (или кодирования), выполняющие преобразование цифровых данных в аналоговый сигнал:

— амплитудная (Amplitude-Shift Keying, ASK),
— частотная (Frequency-Shift Keying, FSK),
— фазовая (Phase-Shift Keying, PSK).

3.3.1. Амплитудная модуляция

При амплитудной модуляции нулевому биту обычно соответствует нулевое значение амплитуды, единичному биту — некоторое, отличное от нуля, значение амплитуды, т. е. представлению нуля или единицы соответствует наличие или отсутствие соответственно несущей частоты при постоянной амплитуде. Результирующий сигнал при этом имеет вид:

$$s(t) = \begin{cases} A\cos(2\pi f_c t), & \text{кодирует двоичную } 1, \\ 0, & \text{кодирует двоичный } 0, \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; t — время.

3.3.2. Частотная модуляция

Частотная модуляция имеет две формы:
- бинарную (Binary FSK, BFSK),
- многочастотную (Multiple FSK, MFSK).

При бинарной частотной модуляции два двоичных числа представляются сигналами двух различных частот, расположенных около несущей. Результирующий сигнал при этом имеет вид:

$$s(t) = \begin{cases} A\cos(2\pi f_1 t), \\ A\cos(2\pi f_2 t), \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_1 и f_2 — частоты, смещённые от несущей частоты f_c на величины, равные по модулю, но противоположные по знаку; t — время.

При многочастотной модуляции для кодирования сигнала используется несколько частот, и за один раз пересылается более одного бита. Сигнал при этом имеет вид:

$$s_i = A\cos(2\pi f_i t), \quad 1 \ll i \ll M,$$
$$f_i = f_c + (2i - 1 - M) f_d, \quad M = 2^L,$$

где $A\cos(2\pi f_i t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; f_d — разностная частота; M — число различных сигнальных посылок; L — количество бит, переданных за один раз; t — время.

Бинарная частотная модуляция менее восприимчива к ошибкам, чем амплитудная модуляция. Многочастотная модуляция эффективнее бинарной, но и более подвержена ошибкам.

3.3.3. Фазовая модуляция

При фазовой модуляции для представления данных выполняется смещение несущего сигнала.

Фазовая модуляция имеет следующие формы:
- двухуровневую (Binary PSK, BPSK),
- дифференциальную (Differential PSK, DPSK),
- квадратурную (Quadrature Phase-Shift Keying, QPSK),
- многоуровневую (Multiple FSK, MFSK).

При двухуровневой фазовой модуляции для представления двух двоичных цифр используются две фазы. При этом результирующий сигнал (для одного периода передачи бита) имеет вид:

$$s(t) = \begin{cases} A\cos(2\pi f_c t) \\ A\cos(2\pi f_c t + \pi) \end{cases} = \begin{cases} A\cos(2\pi f_c t), & \text{кодирует двоичную 1,} \\ -A\cos(2\pi f_c t), & \text{кодирует двоичный 0,} \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; t — время.

При дифференциальной фазовой модуляции для представления двоичного нуля используется сигнал, фаза которого совпадает с фазой предыдущего сигнала, а для представления двоичной единицы — сигнал с фазой, противоположной фазе предыдущего. Такая схема называется *дифференциальной*, поскольку сдвиг фаз выполняется относительно предыдущего переданного бита, а не относительно какого-то эталонного сигнала.

При квадратурной фазовой модуляции каждой сигнальной посылкой представляется более одного бита, при этом вместо сдвига фазы на π, как в двухуровневой модуляции, используются сдвиги фаз, кратные $\pi/2$:

$$s(t) = \begin{cases} A\cos\left(2\pi f_c t + \dfrac{\pi}{4}\right), & \text{кодирует 11,} \\ A\cos\left(2\pi f_c t + \dfrac{3\pi}{4}\right), & \text{кодирует 10,} \\ A\cos\left(2\pi f_c t + \dfrac{5\pi}{4}\right), & \text{кодирует 00,} \\ A\cos\left(2\pi f_c t + \dfrac{7\pi}{4}\right), & \text{кодирует 01,} \end{cases}$$

где $A\cos(2\pi f_c t)$ — несущий сигнал; A — амплитуда; f_c — несущая частота; t — время.

Схема работы многоуровневой фазовой модуляции аналогична схеме работы квадратурной фазовой модуляции, но в каждый момент времени передаётся по три бита, используется восемь различных углов сдвига фаз, для каждого угла используется несколько амплитуд.

3.3.4. Квадратурная амплитудная модуляция

Схема работы квадратурной амплитудной модуляции совмещает в себе принципы амплитудной и фазовой модуляций. На одной несущей частоте одновременно передаются два различных сигнала, но при этом задействованы две копии несущей частоты, сдвинутые относительно друг друга на 90^o, и обе несущие являются амплитудно-модулированными. В приёмнике эти сигналы демодулируются, а результаты объединяются с целью восстановления исходного двоичного сигнала.

При использовании двухуровневой амплитудной модуляции (2QAM) каждый из двух потоков может находиться в одном из двух состояний, а объединённый поток — в одном из четырёх. При использовании четырёхуровневой модуляции (т.е. четырёх различных уровней амплитуды, 4QAM) объединённый поток будет находиться в одном из 16 состояний. Чем больше число состояний, тем выше скорость передачи данных, возможная при определённой ширине полосы. Но чем

больше число состояний, тем выше потенциальная частота возникновения ошибок вследствие помех или поглощения.

3.3.5. Технология расширенного спектра

Основная идея метода состоит в том, чтобы распределить информационный сигнал по широкой полосе радиодиапазона, что в итоге позволит значительно усложнить подавление или перехват сигнала.

3.3.5.1. Расширение спектра скачкообразной перестройкой частоты (Frequency Hopping Spread Spectrum, FHSS)

Передача ведётся с постоянной сменой несущей в пределах широкого диапазона частот. В результате мощность сигнала распределяется по всему диапазону, а прослушивание какой-то определённой частоты даёт только небольшой шум. Последовательность несущих частот псевдослучайна и известна только передатчику и приёмнику. Попытка подавления сигнала в каком-то узком диапазоне почти не ухудшает сигнал, так как подавляется только небольшая часть информации.

На каждой несущей частоте для передачи дискретной информации применяются стандартные методы модуляции — частотная или фазовая. Для синхронизации приёмника и передатчика в течение некоторого времени передаются синхронизирующие последовательности бит. Несущая частота меняется в соответствии с номерами частотных подканалов, вырабатываемых алгоритмом псевдослучайных чисел. Если частота смены подканалов ниже, чем скорость передачи данных в канале, то такой режим называют *медленным расширением спектра*, в противном случае — *быстрым расширением спектра*. Метод быстрого расширения спектра более устойчив к помехам, т.к. помехи, подавляющие сигнал в определённом подканале, не приводят к потере бита, поскольку его значение повторяется несколько раз в различных частотных подканалах. Метод медленного расширения спектра менее устойчив к помехам, но его проще реализовать.

3.3.5.2. Прямое последовательное расширение спектра (Direct Sequence Spread Spectrum, DSSS)

В методе прямого последовательного расширения спектра, в отличие от метода расширения спектра скачкообразной перестройкой частоты, весь частотный диапазон занимается не за счёт постоянных переключений с частоты на частоту, а за счёт того, что каждый бит информации заменяется последовательностью из N бит, что даёт увеличение тактовой скорости передачи сигналов в N раз и соответствующее расширение в N раз спектра сигнала.

Передача двоичной единицы заменяется передачей расширяющей последовательности. Двоичный нуль кодируется инверсным значением расширяющей последовательности. Количество бит в расширяющей последовательности определяет коэффициент расширения исходного кода. Для кодирования битов результирующего кода может использоваться любой вид модуляции. Чем больше коэффициент расширения, тем шире спектр результирующего сигнала и выше степень подавления помех. Но при этом растёт занимаемый каналом диапазон спектра.

Помехи искажают только определённые частоты спектра сигнала, поэтому приёмник с большой степенью вероятности может правильно распознавать передаваемую информацию.

Метод прямого последовательного расширения спектра в меньшей степени защищён от помех, чем метод быстрого расширения спектра, так как мощные помехи влияют на часть спектра, а значит, и на результат распознавания единиц или нулей.

3.4. Кодирование сигнала

Одной из основных задач физического уровня модели OSI является преобразование данных в электромагнитные сигналы, и наоборот. Переход от электромагнитых импульсов к последовательности бит называют *кодированием сигнала*.

Рассмотрим наиболее распространённые методы кодирования (рис. 3.6).

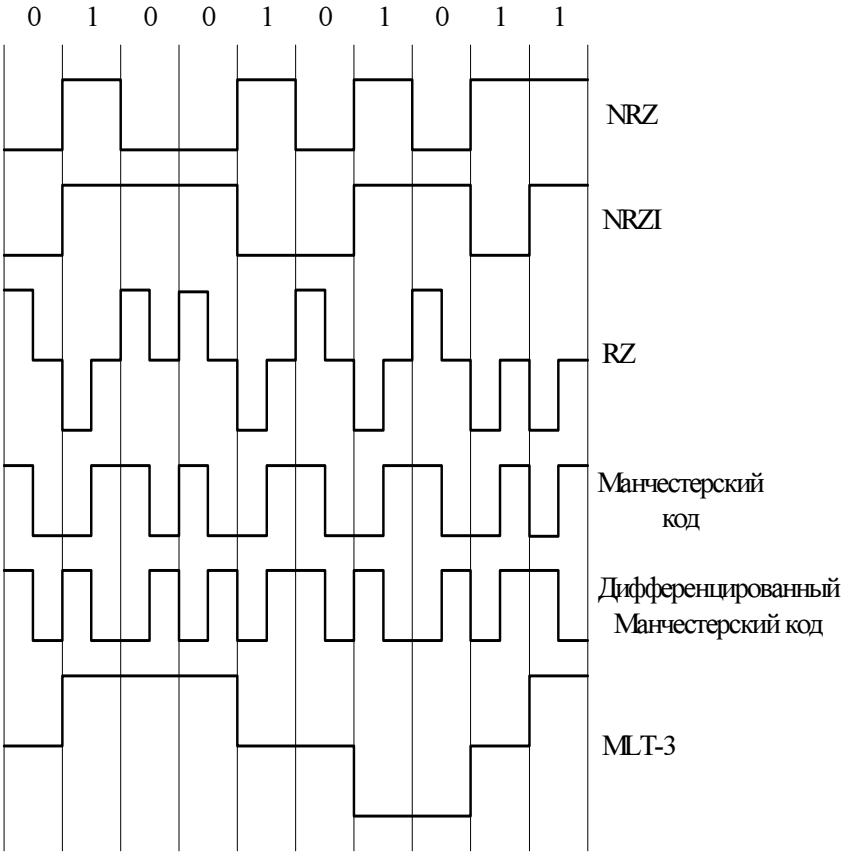


Рис. 3.6. Методы кодирования сигнала

3.4.1. Код NRZ и NRZI

Код *NRZ (Non Return to Zero)* — простейший двухуровневый код. Логической единице соответствует верхний уровень, логическому нулю — нижний, переходы

электрического сигнала происходят на границе битов (рис. 3.7). Код NRZ отличается простотой и обеспечивает высокую скорость передачи, но не имеет синхронизации.

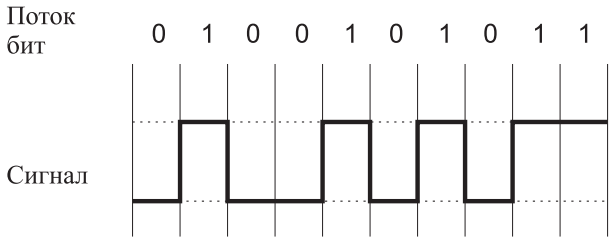


Рис. 3.7. Код NRZ

Код *NRZI (Non Return to Zero Invert to ones)* представляет собой модификацию кода NRZ. В этом двухуровневом коде принимается во внимание значение предыдущего бита. Уровень сигнала меняется, если текущий бит — единица, и повторяет предыдущий, если текущий бит имеет значение 0 (рис. 3.8). NRZI используется в основном для работы с оптоволоконной средой, в сетях 100Base-FX.

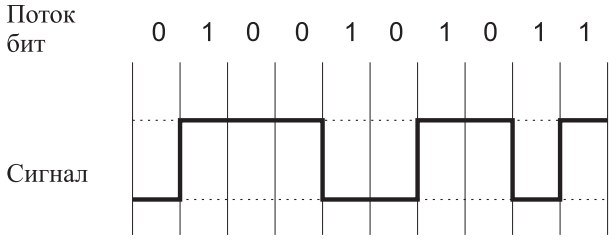


Рис. 3.8. Код NRZI

3.4.2. Код Rz

Код *RZ (Return to Zero)* обеспечивает возвращение к нулю после передачи каждого бита информации. RZ — трёхуровневый код. В центре бита всегда есть переход. Логической единице соответствует отрицательный импульс, логическому нулю — положительный (рис. 3.9). RZ — самосинхронизирующийся код, однако, он не даёт выигрыша в скорости. Код RZ нашёл применение в оптоволоконных сетях.

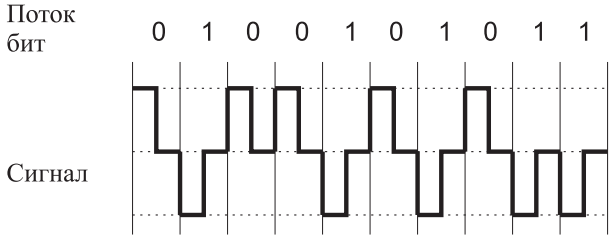


Рис. 3.9. Код Rz

3.4.3. Манчестерский код

Двухуровневый Манчестерский код широко используется в локальных сетях. Логической единице соответствует переход вниз в центре бита, логическому нулю — переход вверх (рис. 3.10). Манчестерский код является самосинхронизирующимся и обладает хорошей помехозащищённостью.

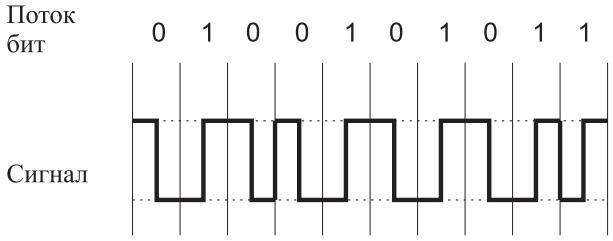


Рис. 3.10. Манчестерский код

3.4.4. Код MLT-3

Код *MLT-3 (Multi Level Transmission-3)* — трёхуровневый код. Как и в NRZI, логической единице соответствует смена уровня сигнала, а при передаче нуля сигнал не меняется (рис. 3.11). Изменение уровня сигнала происходит последовательно с учётом предыдущего перехода. Основной недостаток кода MLT-3 — отсутствие синхронизации. MLT-3 применяется в сетях 100Base-T на основе витой пары.

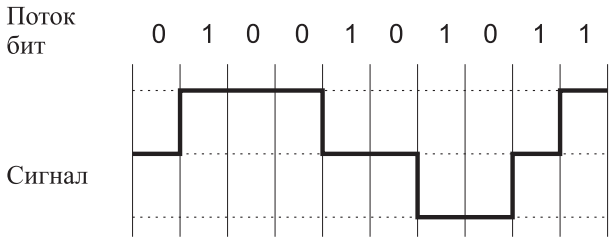


Рис. 3.11. Код MLT-3

3.4.5. Кодирование данных 4B/5B и 8B/6T

В отличие от кодирования сигналов, обеспечивающего переход от импульсов к битам и наоборот, *кодирование данных (физическое кодирование)* преобразует одну последовательность бит в другую.

При использовании несамосинхронизирующихся кодов передача подряд длинной последовательности нулей или единиц приводит к потере несущей и ошибкам приёма. Для исключения таких цепочек и улучшения помехоустойчивости применяется *кодирование данных 4B/5B*, в котором используется 5-битовая основа для передачи 4-битовых сигналов.

При использовании пяти бит для кодирования шестнадцати исходных 4-битовых комбинаций можно построить такую таблицу кодирования, в которой любой

исходный 4-битовый код представляется 5-битовым кодом с чередующимися нулями и единицами, чем обеспечивается синхронизация приёмника и передатчика. Из 32 возможных комбинаций 5-битовых последовательностей (2^5) используются только 16 (2^4), а остальные 16 комбинаций используются в служебных целях (9 комбинаций) или запрещены (7 комбинаций). Так, наличие служебных символов позволяет применять схему непрерывного обмена сигналами между приёмником и передатчиком (например, в спецификациях FX/TX), что даёт возможность контролировать физическое состояние линии и поддерживать синхронность работы приёмника и передатчика. Существование запрещённых комбинаций символов способствует отбраковыванию ошибочных символов. Высокая помехоустойчивость достигается контролем принимаемых данных на 5-битовом интервале.

Таким образом, положительными сторонами кодирования 4B/5B являются синхронизация и улучшение помехоустойчивости. К отрицательным сторонам следует отнести снижение скорости передачи полезной информации (по причине избыточности кода — добавление одного избыточного бита на 4 информационных).

Такая же схема кодирования используется в методах кодирования 8B/10B (8 бит кодируются 10-битным символом) и 5B/6B (5 бит входного потока кодируется 6-битными символами). Кодирование 8B/10B применяется в гигабитных оптоволоконных сетях, поскольку код обеспечивает стабильное соотношение 0 и 1 в выходном потоке, не зависящем от входных данных, что актуально для лазерных оптических передатчиков, так как от данного соотношения зависит их нагрев и при колебании степени нагрева увеличивается количество ошибок приёма. Кодирование 5B/6B применяется в сетевой технологии 100VG-AnyLAN.

Другим способом синхронизации длинных последовательностей нулей или единиц (подобным кодированию 4B/5B) является *кодирование 8B/6T*, которое преобразует 8-битовые последовательности данных в 6 троичных цифр (т.е. цифр, имеющих три состояния). Данный метод кодирования используется в сетях 100Base-T4.

Глава 4. Канальный уровень

4.1. Доступ к среде

При доступе к среде возникает проблема распределения одного широковещательного канала между несколькими пользователями. Можно выделить две схемы выделения канала: *статическую* и *динамическую*. Рассмотрим динамическое выделение канала.

4.1.1. Динамическое выделение канала

Рассмотрим вначале несколько моделей.

1) *Многостанционная модель*. В рамках этой модели рассматривается N независимых станций, каждая из которых порождает кадры с вероятностью $\lambda\Delta t$ за период Δt, где λ — некоторый параметр. После отправки кадра станция блокируется и ничего не предпринимает, пока кадр не будет успешно передан.
2) *Модель единого канала*. Для всех коммуникаций используется один канал. Все станции эквивалентны.
3) *Модель с коллизиями*. Если два кадра передаются одновременно, они перекрываются и возникает коллизия. Все станции могут детектировать коллизии. Эти кадры должны быть переданы ещё раз. За исключением коллизий, других ошибок нет.
4) *Временны́е модели*:
   а) *Модель непрерывного времени*. Передача кадров может произойти в любой момент времени. Время непрерывно.
   б) *Модель тактированного времени*. Время разбивается на дискретные интервалы — *такты (Slots)*. Передача кадров происходит всегда в начальный момент такта.
5) *Модели с несущей*:
   а) *Модель с прослушиванием несущей*. Прежде чем использовать канал, станции запрашивают состояние канала. Если он занят, то все станции перестают его использовать до тех пор, пока он не освободится.
   б) *Модель без прослушивания несущей*. Станции не запрашивают состояние канала перед началом передачи.

4.1.2. Протоколы множественного доступа

Рассмотрим некоторые модели протоколов с множественным доступом.

4.1.2.1. Семейство протоколов ALOHA

В 1970-х гг. в Гавайском университете под руководством Нормана Абрамсона была разработана система ALOHA. Она использовалась для наземной системы радиодоступа.

Центральный узел, называемый базовой станцией, принимает пакеты, передаваемые другими узлами на частоте $f_0 = 417$ МГц и ретранслирует эти пакеты на частоте $f_1 = 413$ МГц. Узлы сети ALOHA передавали пакеты со скоростью 9600 бит/с.

Узлы передают пакеты по общему каналу. Когда передача двух пакетов происходит одновременно, они искажают друг друга. Возникают коллизии. В начальной реализации сети ALOHA центральный узел подтверждает верно принятые пакеты. Когда узел не получает подтверждение за определённый промежуток времени, он полагает, что произошла коллизия, и передаёт пакет снова.

ALOHA не использует контроль несущей и не прекращает передачу пакета при обнаружении конфликта. Контроль несущей бесполезен, поскольку узлы расположены далеко друг от друга, и узел может завершить передачу прежде, чем другой узел заметит передачу. По тем же причинам обнаружение конфликтов слишком запаздывает.

Рассмотрим две версии протокола ALOHA: *чистую (Pure ALOHA)* и *тактированную (синхронную) (Slotted ALOHA)*. В первой используется модель непрерывного времени, а во второй — тактированного.

В модели *Чистая ALOHA* станция начинает передачу данных сразу же, как только у неё появляются данные. При возникновении коллизии посылающая станция ждёт случайный промежуток времени, а затем повторяет передачу этого кадра.

Таким образом, если станция начала передачу в то время, пока предыдущий кадр находится в канале, возникает коллизия. Оба пакета разрушаются и должны быть переданы повторно.

В модели *Тактированная ALOHA* время разбивается на дискретные интервалы. Передача может начаться только в начале такта. Когда у узла появляется новый пакет, он осуществляет его передачу в начале следующего такта. Если в течение этого временного интервала передаётся только один пакет, то передача является успешной. В противном случае возникает коллизия, и узел осуществляет повторную передачу через случайный период времени.

Для реализации тактированной версии протокола ALOHA необходимо приведение узлов к общему эталону времени для определения начала временны́х интервалов.

4.1.2.2. Протоколы множественного доступа с контролем несущей

Протоколы, в которых станции контролируют несущую, называются *протоколами с контролем несущей (Carrier Sense, CS)*.

Рассмотрим несколько видов протоколов семейства *CSMA (Carrier Sense Multiple Access)*.

1-устойчивый (1-persistent) CSMA. Когда станция готова к передаче данных, она прослушивает канал, чтобы определить, не передаёт ли данные кто-либо другой. Если канал занят, станция ждёт, когда он освободится. Если же канал свободен, станция передаёт информацию. При возникновении коллизии станция ждёт случайный промежуток времени, а потом продолжает действовать по вышеописанному алгоритму. Протокол называется 1-устойчивый, потому что в случае свободного канала станция осуществляет передачу с вероятностью 1.

Неустойчивый (nonpersistent) CSMA. Этот случай немного отличается от предыдущего. Здесь опять перед передачей данных станция прослушивает канал.

Но в случае, когда канал уже используется, станция ожидает случайный период времени и повторяет алгоритм.

p-устойчивый (p-persistent) CSMA. Данный вид применяется к тактированному каналу. Если канал свободен, то передача осуществляется с вероятностью p. Соответственно с вероятностью $q = 1 - p$ станция будет ждать следующего такта. Если канал свободен, то передача данных или ожидание следующего такта происходят с вероятностью p и q соответственно. Этот процесс продолжается до тех пор, пока либо кадр не будет передан, либо другая станция не начнёт передачу. В последнем случае возникает коллизия; станция ожидает случайный период времени и пытается снова осуществить передачу данных. Если же при прослушивании канала он оказывается занятым, то станция ждёт до начала следующего такта и повторяет алгоритм.

Устойчивый и неустойчивый CSMA являются непосредственным улучшением протоколов семейства ALOHA, поскольку в них для определения состояния канала осуществляется его прослушивание.

Протоколы множественного доступа с контролем несущей с определением коллизий (Carrier Sense Multiple Access with Collision Detection, CSMA/CD) являются дальнейшим улучшение протоколов, рассмотренных в предыдущем пункте. Здесь при возникновении коллизии станции сразу же прекращают передачу данных (вместо того, чтобы продолжать передачу, что бессмысленно). Это позволяет сэкономить и время, и полосу пропускания.

При обнаружении коллизии станция прекращает передачу данных и ждёт случайное время. По истечении данного времени станция опять пытается передать данные.

Разберём алгоритм определения коллизий подробнее. Пусть две станции начали передачу данных в один и тот же момент времени. Время определения коллизии будет зависеть от времени распространения сигнала между двумя этими станциями.

Обозначим через τ время прохождения сигнала между двумя наиболее удалёнными друг от друга станциями. Пусть первая станция начинает передачу данных в некоторый момент времени t_0. Если в промежуток времени $[t_0, t_0 + \tau)$ вторая станция тоже начнёт передачу, то она обнаружит коллизию. Чтобы коллизию обнаружила и первая станция, сигнал должен вернуться обратно, то есть коллизия будет обнаружена в промежуток времени $[t_0, t_0 + \tau)$. Временной интервал 2τ называется *временем двойного оборота (Path Delay Value, PDV)*.

4.2. Группа стандартов IEEE 802

Как было сказано в разделе 2.2.3, стандарты IEEE 802 охватывают только два нижних уровня семиуровневой эталонной модели ISO/OSI — физический и канальный (рис. 4.1), отражая тем самым специфику локальных сетей.

Многие сетевые стандарты IEEE легли в основу сетевых стандартов ISO и IEC. В 1980 г. в IEEE был организован комитет 802 по стандартизации локальных сетей. Результатом его деятельности стала разработка семейства протоколов IEEE 802, содержащих рекомендации по проектированию нижних уровней локальных сетей. Эти стандарты были созданы на основе распространённых фирменных стандартов сетей Arcnet, Ethernet, Token Ring.

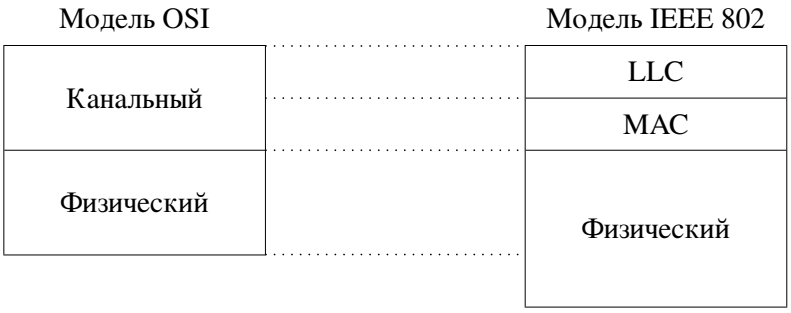


Рис. 4.1. Соответствие эталонных моделей ISO/OSI и IEEE 802

4.2.1. Структура стандартов IEEE 802

Нумерация стандартов IEEE из серии 802 производится в соответствии со своей собственной схемой. Если за цифрой следует прописная буква, то это отдельный стандарт, если же за цифрой следует строчная буква, то это дополнение к стандарту или часть стандарта, обозначаемого несколькими цифрами.

Стандарты, разрабатываемые подкомитетом 802.1, носят общий для всех технологий характер. Именно в нём были разработаны общие определения локальных сетей и их свойств, обозначена связь эталонных моделей IEEE 802 и ISO/OSI. Также сюда входят стандарты *межсетевого (internetworking) взаимодействия*, описывающие взаимодействие разных технологий, и стандарты построения более сложных сетей. Это, например, стандарт IEEE 802.1D, описывающий логику работы моста (коммутатора), стандарт IEEE 802.1Q, определяющий способ построения виртуальных локальных сетей (Virtual Local Area Network, VLAN) в сетях на основе коммутаторов.

Включение уровня LLC в стандарт IEEE позволило определить стандартный интерфейс на уровне MAC, однако используют настоящий LLC (т.е. LLC Type 2) только два протокола — SNA (Systems Network Architecture) и NetBEUI (NetBIOS Extended User Interface, расширенный пользовательский интерфейс NetBIOS), называемый также NetBIOS поверх LLC. Обычно применяются только заголовки LLC Type 1 в качестве заглушки (stub) для протоколов верхнего уровня. Стандартом LLC занимается подкомитет 802.2.

Стандарты 802.3, 802.4, 802.5 описывают технологии, созданные на основе фирменных технологий. Основу стандарта IEEE 802.3 составила технология Ethernet, разработанная компаниями DEC, Intel и Xerox. Стандарт IEEE 802.4 создан на основе технологии ArcNet фирмы Datapoint Corporation. Стандарт IEEE 802.5 базируется на технологии Token Ring компании IBM.

Исходные фирменные технологии и стандарты IEEE 802 в ряде случаев довольно долго существовали параллельно. Технология ArcNet так до конца и не была приведена в соответствие со стандартом IEEE 802.4[1]. Из-за того, что IBM регулярно вносит усовершенствования в технологию Token Ring, периодически возникают расхождения между стандартом IEEE 802.5 и данной технологией.

Сегодня комитет IEEE 802 включает следующие подкомитеты [5]:

802.1 Internetworking — объединение сетей.

802.1B Стандарт управления локальными/региональными сетями. Одобренный в 1992 г., он вместе с 802.1k лёг в основу ISO/IEC 15802-2.

[1] Примерно в 1993 г. производство оборудования ArcNet прекращено.

802.1D Стандарт межсоединения локальных сетей с помощью мостов уровня MAC. Одобренный в 1990 г., он лёг в основу ISO/IEC 10038.

802.1E Стандарт на протоколы системной нагрузки для локальных и региональных сетей. Одобренный в 1990 г., он лёг в основу ISO/IEC 10038.

802.1F Стандарт определения управляющей информации для серии 802; одобрен в 1993 г.

802.1g Предложение по стандарту на удалённые мосты уровня MAC.

802.1H Рекомендуемые правила организации мостов MAC в сетях Ethernet 2.0; одобрены в 1995 г.

802.1i Стандарт на использование FDDI в качестве моста уровня MAC; одобрен в 1992 г. и включён в ISO/IEC 10038.

802.1j Дополнение к 802.1D; одобрено в 1996 г. Данный стандарт описывает связь локальных сетей с помощью мостов уровня MAC.

802.1k Стандарт для локальных и региональных сетей на обнаружение и динамический контроль маршрутизации событий; одобрен в 1993 г. и вместе с 802.1B лёг в основу ISO/IEC 15802-2.

802.1m Описание соответствий для 802.1E, рассматривающее определения и правила управляемых объектов для протокола системной нагрузки; одобрено в 1993 году и включено в ISO/IEC 15802-4.

802.1p Предложение по стандарту для локальных и региональных сетей, касающееся ускорения обработки трафика и многоадресной фильтрации с помощью мостов уровня MAC.

802.1Q Предложение по стандарту на виртуальные локальные сети с мостами.

802.2 Logical Link Control, LLC — управление логической передачей данных. Стандарт для логического управления каналом связи локальных и региональных сетей, в основном с помощью мостов; лёг в основу ISO/IEC 8802-2. Текущая версия, одобренная в 1994 г., заменила более ранний стандарт 802.2 от 1989 г.

802.3 Стандарт на метод коллективного доступа для локальных сетей CSMA/CD и на физический уровень. Он положен в основу ISO/IEC 8802-3. Также его называют стандартом Ethernet.

802.3b Стандарт на устройства подключения к широкополосной среде передачи для 10Broad36. Одобрен в 1985 г. и включён в ISO/IEC 8802-3.

802.3c Стандарт на повторители в сети с немодулированной передачей на 10 Мбит/с. Одобрен в 1985 г. и включён в ISO/IEC 8802-3.

802.3d Стандарт на устройства подключения к среде передачи и спецификации среды с немодулированной передачей для каналов с оптическими повторителями. Одобрен в 1987 г. и включён в ISO/IEC 8802-3.

802.3e Стандарт на сигнализацию на физическом уровне, подключение к среде передачи и спецификации на среду с немодулированной передачей для сети на 1 Мбит/с, иными словами, 1Base5. Одобрен в 1987 году и включён в ISO/IEC 8802-3.

802.3h Стандарт на управление уровнем в сетях коллективного доступа CSMA/CD. Одобрен в 1990 г. и включён в ISO/IEC 8802-3.

802.3i Стандарт охватывает две области: многосегментную сеть немодулированной передачи на 10 Мбит/с и витую пару для сети 10BaseT. Одобрен в 1990 г. и включён в ISO/IEC 8802-3.

802.3j Стандарт на активные и пассивные сегменты в топологии звезда на 10 Мбит/с с использованием оптической среды передачи, т.е. 10BaseF. Одобрен в 1993 г. и включён в ISO/IEC 8802-3.

802.3k Стандарт на управление уровнем для повторителей в сети с немодулированной передачей на 10 Мбит/с. Одобрен в 1992 г. и включён в ISO/IEC 8802-3.

802.3l Описание соответствия для протоколов устройств подключения к среде передачи 10BaseT. Одобрено в 1992 г. и включено в ISO/IEC 8802-3.

802.3p Стандарт на управление уровнем для устройств подключения к среде с немодулированной передачей на 10 Мбит/с. Одобрен в 1993 г. и включён в ISO/IEC 8802-3.

802.3q Рекомендации по разработке управляемых объектов. Одобрены в 1993 г. и включены в ISO/IEC 8802-3.

802.3r Стандарт на метод коллективного доступа к среде передачи CSMA/CD, а также спецификации физического уровня для 10Base5. Пересмотрен в 1996 г.

802.3t Стандарт на поддержку 120-омных кабелей в сегментах с симплексными каналами 10BaseT. Включён в ISO/IEC 8802-3, одобрен в 1995 г.

802.3u Дополнение к 802.3, касающееся параметров MAC, физического уровня и повторителей на 100 Мбит/с, т. е. 100BaseT или, иначе, Fast Ethernet. Одобрено в 1995 г.

802.3v Стандарт для поддержки 150-омных кабелей в сегментах с каналами 10BaseT. Одобрен в 1995 г. и включён в ISO/IEC 8802-3.

802.3w Предложение по стандарту на усовершенствованные алгоритмы MAC.

802.3x Предложение по стандарту на полнодуплексный режим для 802.3.

802.3y Предложение по спецификации физического уровня для работы на 100 Мбит/с по двум парам категории 3 или ещё лучше сбалансированного кабеля на основе витой пары, т.е. 100BaseT2.

802.3z Предложение по стандарту на физический уровень, повторители и управляющие параметры для работы на 1000 Мбит/с, часто называемое Gigabit Ethernet.

802.4 Token Bus LAN. Стандарт на метод доступа к шине с передачей маркера и спецификации физического уровня. Одобрен в 1990 г.

802.5 Стандарт на методы доступа к кольцу с передачей маркера и спецификации физического уровня, т.е. на общую архитектуру Token Ring. Лёг в основу ISO/IEC 8802-5, текущая версия была одобрена в 1995 г.

802.6 Семейство стандартов на сеть с двойной шиной и распределённой очередью. Одобрено в 1990 г.

802.7 Broadband Technical Advisory Group — техническая консультационная группа по широкополосной передаче.

802.8 Fiber Optic Technical Advisory Group — техническая консультационная группа по волоконно-оптическим сетям.

802.9 Integrated Voice and Data Networks — интегрированные сети передачи данных и голоса. Стандарт на локальную сеть с интеграцией услуг (Integrated Services LAN) для подключения локальных сетей 802.x к общедоступным и частным магистральным сетям, таким как FDDI и ISDN. Одобрен в 1994 году и лёг в основу ISO/IEC 8802-9.

802.10 Стандарт на защиту локальных сетей Interoperable LAN Security, известный так же, как SILS. Одобрен в 1992 г.

802.11 Стандарт на беспроводные локальные сети (Wireless Local Area Networks, WLAN). Стандарт на уровень MAC и спецификации физического уровня для беспроводных локальных сетей. Предлагаемый проект рассчитан на диапазон 2,4 ГГц.

- **802.11a** Редакция стандарта 802.11 IEEE, в которой рассматриваются сети, работающие со скоростями до 54 Мбит/с по технологии DSSS (Direct Sequence Spread Spectrum).
- **802.11b** Редакция стандарта 802.11 IEEE, в которой рассматриваются сети, работающие со скоростями до 11 Мбит/с по технологии DSSS (Direct Sequence Spread Spectrum).
- **802.11e** Редакция стандарта 802.11 IEEE по качеству услуг (Quality of Service, QoS).
- **802.11f** Редакция стандарта 802.11 IEEE, определяющая протокол взаимодействия точек доступа (Inter-Access Point Protocol).
- **802.11g** Редакция стандарта 802.11 IEEE, в которой рассматриваются сети, работающие со скоростями до 54 Мбит/с по технологии DSSS, обратно совместимые со стандартом 802.11b.
- **802.11h** Редакция стандарта 802.11 IEEE, определяющая управляемый спектр для 802.11a (Managed Spectrum for 802.11a).
- **802.11i** Стандарт IEEE, относящийся к безопасности беспроводных сетей. В нём объединены протоколы 802.1x и TKIP/CCMP, что позволяет обеспечить аутентификацию пользователей, конфиденциальность и целостность данных в беспроводных локальных сетях.

802.12 Demand Priority Access LAN, 100VG-AnyLAN — локальные сети с методом доступа по требованию с приоритетами.

802.15 Стандарт беспроводных персональных сетей (Wireless Personal Area Networks, WPAN), работающих на ограниченных расстояниях.

- **802.15.1** Bluetooth (базируется на спецификациях Bluetooth v1.x).
- **802.15.3** Стандарт беспроводных сетей, являющийся прямым наследником Bluetooth (частота 2,4 ГГц). Определяет беспроводные персональные сети со скоростью передачи данных 55 Мбит/с для мультимедийных приложений.
- **802.15.4 (ZigBee)** Стандарт определяет спецификации физического слоя и протокол управления доступом (MAC), предлагая поддержку различных топологий сетей.
- **802.15.4a** Технология сверхширокополосной связи (Ultra Wideband, UWB), при помощи которой можно создавать специальные сети, где несколько сверхширокополосных устройств смогут поддерживать связь между любыми узлами.

802.16 Стандарт широкополосной беспроводной связи (Broadband Wireless Access, BWA), в частности, Worldwide Interoperability for Microwave Access — WiMax.
802.17 Стандарт на адаптивные, кольцевые, высокоскоростные сети.
802.18 Техническая консультативная группа по регулированию радиотехнологий (Radio Regulatory Technical Advisory Group, RRTAG).
802.19 Техническая консультативная группа по совместимости (Coexistence Technical Advisory Group, CoTAG).
802.20 Стандарт на мобильный широкополосный беспроводной доступ (Mobile Broadband Wireless Access, MBWA).
802.21 Стандарт на услуги эстафетной передачи соединения независимо от среды (Media Independent Handover Services, MIHS).
802.22 Стандарт на беспроводные региональные сети (Wireless Regional Area Networks, WRAN).

4.2.2. Протокол MAC

4.2.2.1. Адресация MAC-уровня

Протоколы семейства IEEE 802 используют 48-битную схему адресации MAC-уровня (рис. 4.2). IEEE также предлагал 16-разрядный MAC-адрес, но он не получил большого распространения.

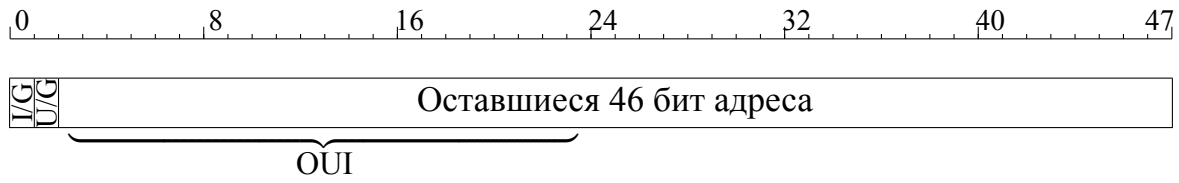


Рис. 4.2. Структура MAC-адреса IEEE. I/G: $= 0$ — индивидуальный адрес, $= 1$ — групповой адрес; U/G: $= 0$ — глобально администрируемый адрес, $= 1$ — локально администрируемый адрес

Первый бит MAC-адреса получателя называется *индивидуальным/групповым битом (Individual/Group, I/G)*. Если он установлен в 0, то кадр послан определённой рабочей станции, если же он установлен в 1, то кадр является широковещательным (поэтому данный бит также называют *широковещательным битом*). Если и все остальные биты адреса установлены в 1, то широковещательный кадр предназначен всем станциям, в противном случае мы имеем дело с *групповой (multicast)* рассылкой кадра на выделенное подмножество станций (станции должны быть сконфигурированы для приёма групповых адресов).

В адресе источника первый бит называется *индикатором маршрута от источника (Source Route Indicator)*.

Три старших байта адреса называют *зашитым адресом (Burned in Address, BIA)* или *уникальным идентификатором организации (Organizationally Unique Identifier, OUI)*. Этот идентификатор выдаётся каждому производителю оборудования (распределением OUI занимался сначала Xerox, теперь эти полномочия

делегированы IEEE). За уникальность младших трёх байт адреса отвечает сам производитель.

Второй бит адреса определяет способ назначения адреса. Если он выставлен в 0, то адрес является *централизованно или глобально администрируемым (Universally/Globally Administered)*. В этом случае сохраняется адрес, заданный производителем. Если же этот бит установлен в 1, то адрес является *локально администрируемым (Locally Administered Address, LAA)*, т.е. текущий адрес заменяет адрес, установленный производителем.

IBM ввела локально администрируемые адреса, чтобы пользователи могли работать с адресом сети SNA при обращении извне к этой сети. Очевидный недостаток этих адресов — возможность появления в сети дублированных адресов. Локально администрируемые адреса допустимы и в Ethernet.

Следует отметить специфику MAC-адреса для Ethernet (рис. 4.3). В стандарте Ethernet младший бит байта отображается в самой левой позиции, а старший бит — в самой правой. При этом порядок следования байтов остаётся традиционным. Обычно же младшим считается самый правый бит байта, а старшим — самый левый.

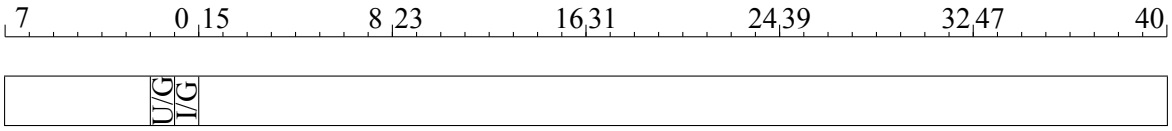


Рис. 4.3. Структура MAC-адреса Ethernet

4.2.3. Протокол IEEE 802.2 LLC

Уровень управления логическим каналом (Logical Link Control, LLC) отвечает за передачу кадров между узлами с различной степенью надёжности, а также реализует функции интерфейса с прилегающим к нему сетевым уровнем.

Протоколы сетевого уровня передают данные для протокола LLC: свой пакет, адресную информацию об узле назначения, требования к качеству транспортных услуг, которое должен обеспечить протокол LLC. Протокол LLC помещает пакет протокола верхнего уровня в свой кадр, дополняя необходимыми служебными полями. Потом протокол LLC передаёт свой кадр соответствующему протоколу уровня MAC.

В основу протокола LLC положен протокол *HDLC (Hight-level Data Link Control Procedure)*. Поскольку данный протокол пришлось сопрягать с разными фирменными протоколами, то на уровне LLC пришлось ввести три типа процедур, к одной из которых может обращаться протокол сетевого уровня.

В соответствии со стандартом IEEE 802.2 уровень LLC предоставляет верхним уровням три типа процедур:

- *LLC1, Type 1, Conectionless* — процедура без установления соединения и без подтверждения;
- *LLC2, Type 2, Connection-Oriented* — процедура с установлением соединения и с подтверждением;

— *LLC3, Type 3* — процедура без установления соединения, но с подтверждением.

Процедура без установления соединения и без подтверждения LLC1 представляет собой заглушку (stub) для мультиплексирования или идентифицирует протокол следующего уровня. Она позволяет передавать данные с минимумом издержек. Это датаграммный режим работы. В этом режиме работают такие стеки протоколов, как TCP/IP, IPX/SPX.

Процедура с установлением соединения и с подтверждением LLC2 предоставляет функции транспортного уровня на уровне DLC без участия промежуточного сетевого уровня. Она даёт возможность установить логическое соединение перед началом передачи блока данных и выполнить процедуры восстановления после ошибок и упорядочивания потока этих блоков в рамках установленного соединения.

Протокол LLC Type 2 применяется на сегодняшний день только в двух случаях:

— в стеке протоколов SNA, когда на нижнем уровне используется Token Ring;
— в NetBEUI (NetBIOS поверх LLC).

Процедура без установления соединения, но с подтверждением LLC3 используется в случае, когда временны́е издержки установления логического соединения перед отправкой данных не приемлемы, а подтверждение приёма данных необходимо (например, при использовании сетей в системах реального времени).

По своему назначению все кадры уровня LLC подразделяются на три типа.

— *Информационные кадры (Information, I-frame)* предназначены для передачи информации в процедурах с установлением логического соединения LLC2 и должны обязательно содержать поле информации. В процессе передачи информационных блоков осуществляется их нумерация в режиме скользящего окна.
— *Управляющие кадры (Supervisory, S-frame)* предназначены для передачи команд и ответов в процедурах с установлением логического соединения LLC2, в том числе и запросов на повторную передачу искажённых информационных блоков.
— *Ненумерованные кадры (Unnumbered, U-frame)* предназначены для передачи ненумерованных команд и ответов, выполняющих в процедурах без установления логического соединения передачу информации, идентификацию и тестирование LLC-уровня, а в процедурах с установлением логического соединения LLC2 — установление и разъединение логического соединения, а также информирование об ошибках.

Все типы кадров LLC имеют единый формат (рис. 4.4). Кадр обрамляется двумя однобайтовыми полями *Флаг*, содержащими значение 0b01111110. Флаги используются для определения границ кадра LLC. При вложении кадра LLC в кадр MAC флаги отбрасываются.

Рис. 4.4. Формат кадра LLC

Заголовок кадра LLC состоит из трёх полей:
— точки доступа к службе получателя (Destination Service Access Point, DSAP);
— точки доступа к службе источника (Source Service Access Point, SSAP);
— управляющего поля (Control).

В *поле данных* вкладываются пакеты протоколов вышележащих уровней. Поле данных может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.

Поля DSAP и SSAP имеют размер 1 байт каждое (рис. 4.5). Они служат для идентификации протокола верхнего уровня, инкапсулировавшего данные в кадр LLC. Служба может иметь несколько SAP, что может быть использовано протоколом узла отправителя в специальных целях, например, для уведомления узла-получателя о переходе протокола-отправителя в некий специфический режим работы.

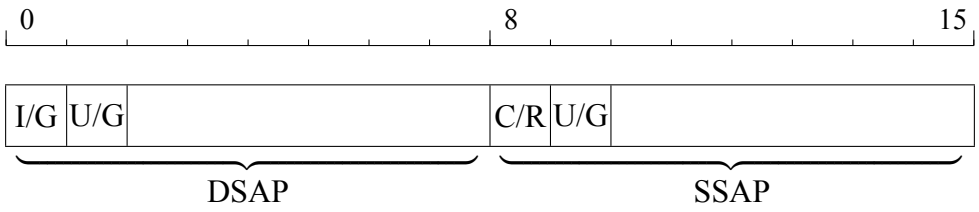


Рис. 4.5. Структура полей SAP. U/G: $= 0$ — глобально администрируемая точка доступа к службе, $= 1$ — локально администрируемая точка доступа к службе; I/G: $= 0$ — индивидуальная точка доступа к службе, $= 1$ — групповая точка доступа к службе; C/R: $= 0$ — команда, $= 1$ — отклик

Поле управления длиной 1 или 2 байта имеет разную структуру в зависимости от типа кадра LLC (рис. 4.6).

В LLC1 используется только один тип кадра — *ненумерованный* (рис. 4.6в). У этого кадра поле управления имеет длину 1 байт. Все подполя поля управления ненумерованных кадров в этом режиме имеют нулевые значения, так что значимыми остаются только два первых бита, используемые как признак типа кадра.

В LLC2 используются все три типа кадров. В этом режиме кадры делятся на команды и ответы на эти команды. Бит *P/F (Poll/Final — опрос/завершение)* имеет следующее значение: в командах он называется битом *Poll* и требует, чтобы на команду был дан ответ, а в ответах он называется битом *Final* и говорит о том, что ответ состоит из одного кадра.

Ненумерованные кадры используются на начальной стадии взаимодействия двух узлов — стадии установления соединения по протоколу LLC2. Поле *M* ненумерованных кадров определяет несколько типов команд, которыми пользуются два узла на этапе установления соединением, например:
— *установка сбалансированного асинхронного расширенного режима (Set Asynchronouse Balanced Mode Extended, SABME)*, что является запросом на установление соединением, причём расширенный режим обозначает использование двухбайтных полей управления для кадров остальных двух типов;
— *ненумерованное подтверждение (UA)* служит для подтверждения установления или разрыва соединения;

— *сброс соединения (REST)* является запросом на разрыв соединения.

Рис. 4.6. Структура поля управления кадров LLC: P/F — бит опрос/завершение; N(S) — порядковый номер отправки; N(R) — порядковый номер получения

После установления соединения данные начинают передаваться в *информационных кадрах* (рис. 4.6а). Логический канал протокола LLC2 является дуплексным. В дуплексном режиме положительные квитанции на кадры также передаются в *информационных кадрах*.

Управляющие кадры (рис. 4.6б) используются для передачи отрицательных квитанций или в полудуплексном режиме (когда нет потока кадров в обратном направлении).

В состав управляющих кадров входят следующие:

— отказ (Reject, REJ);
— приёмник не готов (Receiver Not Ready, RNR);
— приёмник готов (Receiver Ready, RR).

Команда *RR* часто используется как положительная квитанция в случае, если поток данных от приёмника к передатчику отсутствует, а команда *RNR* — для замедления потока данных к приёмнику. Это может быть необходимо, когда приёмник не успевает обработать поток кадров. Получение кадра *RNR* требует от отправителя полной остановки передачи до получения кадра *RR*. С помощью этих команд осуществляется управление потоком данных.

Поле *N(S)* указывает номер отправленного кадра, а поле *N(R)* — номер кадра, который приёмник ожидает получить от передатчика следующим. Поскольку длина каждого из этих полей равна 7 бит, то они могут принимать значения в диапазоне от 0 до 127.

Подтверждение отсылается получателем отправителю с порядковым номером, равным номеру следующего кадра, который ожидает принять получатель от

отправителя. Если же приёмник получает от отправителя кадр с номером, не равным ожидаемому, то этот кадр отбрасывается и посылается отрицательная квитанция *REJ* с номером ожидаемого кадра. При получении отрицательной квитанции передатчик обязан повторить передачу кадра с требуемым номером, а также всех кадров с большими номерами, которые он уже успел отослать.

При работе протокола LLC2 используется скользящее окно размером в 127 кадров.

4.3. Технология Ethernet

4.3.1. Метод доступа CSMA/CD

Метод доступа к среде в технологии Ethernet является вариантом метода CSMA/CD (см. раздел 4.1.2.2), а именно метод CSMA/CD *с двоичной экспоненциальной отсрочкой (Binary Exponential Backoff)*.

Если станция готова к передаче данных, то она действует по следующему алгоритму:

1) Станция ожидает освобождение канала.
2) После освобождения канала перед непосредственной передачей станция выдерживает паузу, называемую *межкадровым интервалом (Inter Packet Gap, IPG)*, длительность которой равна времени передачи 96 бит. Для скорости 10 Мбит/с пауза составляет 9,6 мкс, а для скорости 100 Мбит/с — 0,96 мкс. Эта пауза нужна для предотвращения монопольного захвата сети одной станцией.

   Во время передачи станция продолжает контролировать состояние канала. Если передаваемый и наблюдаемый сигналы отличаются, то считается, что обнаружена коллизия.
3) Если конфликт выявляется во время передачи преамбулы (подробнее структура кадра Ethernet будет рассмотрена в разделе 4.3.2), то оставшаяся часть преамбулы всё равно передаётся, чтобы усилить сигнал коллизии. Если конфликт возникает во время пересылки остальной части кадра, станция пересылает последовательность из 32 бит, называемую *jam-последовательностью*.
4) После прекращения передачи пакета станция ожидает случайное время, затем переходит к шагу 1.

Рассмотрим *алгоритм выбора случайного времени ожидания*. После возникновения коллизии время разбивается на дискретные промежутки, длительность каждого из которых устанавливается равной 512 bt[1]. Назовём этот промежуток *интервалом отсрочки*.

После первой коллизии станции ожидают 0 или 1 интервал отсрочки. После второй период ожидания длится 0, 1, 2 или 3 интервала отсрочки. Иными словами, выбирается количество интервалов отсрочки из интервала $[0, 2^n - 1]$, где n — номер попытки. После десятой попытки верхняя граница интервала фиксируется. После шестнадцатой попытки передатчик должен прекратить передачу и отбросить этот кадр.

[1] Битовый интервал (bit time, bt) — время между появлением двух последовательных бит данных на кабеле. Например, для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс.

4.3.2. Форматы кадров Ethernet

В процессе развития Ethernet и стандарта IEEE 802.3 было предложено 4 варианта формата кадра. В 1980 г. консорциум трёх фирм DEC, Intel, Xerox представил на рассмотрение комитета 802.3 свою версию стандарта Ethernet (тип кадра Ethernet DIX), но комитет принял стандарт, отличающийся деталями (в том числе и форматом кадра) от предложения DIX (тип кадра 802.3/LLC). Novell, являющаяся в то время лидером сетевой индустрии в области персональных компьютеров, предложила свой формат кадра (Raw 802.3). Четвёртый вариант был предложен комитетом 802.2 для ликвидации недостатков формата кадра 802.3/LLC и приведения всех форматов кадров к общему виду (тип кадра Ethernet SNAP).

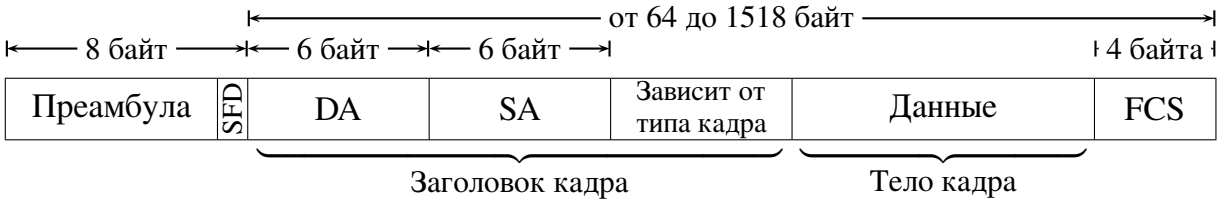


Рис. 4.7. Формат кадра Ethernet

Каждый кадр начинается с *преамбулы (Preamble)* (длина 7 байт), заполненной шаблоном 0b10101010 (для синхронизации источника и получателя). После преамбулы идёт байт *начального ограничителя кадра (Start of Frame Delimiter, SFD)*, содержащий последовательность 0b10101011 и указывающий на начало собственно кадра.

Далее идут поля *адрес получателя (Destination Address, DA)* и *адрес отправителя (Source Address, SA)*. В Ethernet используют 48-битные адреса MAC-уровня IEEE (см. раздел 4.2.2.1).

Следующее поле имеет разный смысл и разную длину в зависимости от типа кадра:

— Тип кадра Ethernet DIX.
  Тип кадра Ethernet DIX — изначальный тип кадра стандарта Ethernet. Этот тип кадра носит также названия Ethertype, Ethernet II (в терминологии NetWare). После поля адреса источника этот тип кадра содержит 16-битное поле *типа (Ethertype)*, идентифицирующее инкапсулированный в кадре протокол верхнего уровня (рис. 4.8а).
— Тип кадра Raw 802.3.
  Этот тип кадра предложен компанией Novell для своей системы NetWare. Он также носит названия Novell 802.3, Ethernet 802.3 (в терминологии NetWare).
  За адресом источника он содержит 16-битное поле *длины (Length, L)*, определяющее число байт, следующее за полем длины (без учёта поля контрольной суммы) (рис. 4.8б).
  В этот тип кадра всегда вкладывается пакет протокола IPX. Первые два байта заголовка протокола IPX содержат контрольную сумму датаграммы IPX. Однако по умолчанию это поле не используется и выставлено в 0xFFFF.
— Тип кадра 802.3/LLC.
  Поскольку группа стандартов IEEE 802 разделяет канальный уровень на подуровни MAC и LLC, то в кадр MAC-подуровня вкладывается кадр LLC-подуровня (см. раздел 4.2.3).

За полем адреса источника идёт 16-битное поле *длины (Length, L)*, определяющее число байт, следующее за полем длины (без учёта поля контрольной суммы).

За ним следует заголовок LLC (рис. 4.8в). Он состоит из 8-разрядных полей *точки доступа к услуге источника (Source Service Access Point, SSAP)* и *точки доступа к услуге получателя (Destination Service Access Point, DSAP)*, а также поля управления, имеющего длину 8 или 16 бит, в зависимости от типа протокола LLC.

Поля SSAP и DSAP размером по 6 бит (см. рис. 4.5) предназначены для описания типа протокола следующего уровня. Но при такой разрядности можно указать не более 64 различных протоколов. Таким образом, недостаточный размер полей SAP создаёт трудности при применении этого типа кадра (например, нет типа SAP для протокола ARP).

— Тип кадра Ethernet SNAP.

Преждевременная стандартизация протокола LLC привела к значительным трудностям в применении типа кадра 802.3/LLC. Для решения этой проблемы комитетом 802.2 был предложен тип кадра *Ethernet SNAP (SubNetwork Access Protocol, протокол доступа к подсети)*.

Кадр Ethernet SNAP является расширением кадра 802.3/LLC за счёт введения дополнительного заголовка протокола SNAP. Заголовок SNAP состоит из 3-байтного поля *уникального идентификатора организации (Organizationally Unique Identifier, OUI)* и 2-байтного поля *типа (Type, Ethertype)*. Тип идентифицирует протокол верхнего уровня, а поле OUI определяет идентификатор организации, контролирующей назначение кодов типа протокола. Коды протоколов для стандартов IEEE 802 контролирует IEEE, имеющая код OUI, равный 0x000000. Для этого кода OUI поле типа для Ethernet SNAP совпадает со значением типа для Ethernet DIX.

Протокол SNAP вкладывается в протокол LLC1. Код SAP для него — 0xAA. Поле управления устанавливается в 0x03, что соответствует использованию ненумерованных кадров (см. рис. 4.6в, учитывая обратный порядок записи битов в байте в протоколе Ethernet).

Далее идёт поле *данных (Data)*. Если длина поля данных недостаточна для получения минимальной длины кадра, то вводится дополнительное поле *заполнения (Padding)*, призванное обеспечить минимальную длину кадра.

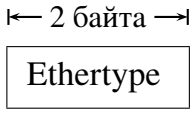
В конце кадра идёт 32-битное поле *контрольной суммы (Frame Check Sequence, FCS)*. Контрольная сумма вычисляется по алгоритму CRC-32.

Размер кадра Ethernet от 64 до 1518 байт (без учёта преамбулы, но с учётом поля контрольной суммы) (см. рис. 4.7).

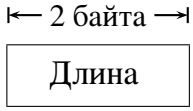
Алгоритм автоматического распознавания разных типов кадров Ethernet достаточно прост.

Поле, следующее за полем адреса источника, имеет длину 2 байта и может быть либо полем Ethertype, либо полем длины данных. Максимальная длина поля данных равна 1500 байт (0x05DC). Значение поля Ethertype всегда больше, чем 0x05DC. Следовательно, если значение поля больше, чем 0x05DC, то мы имеем кадр Ethernet DIX. В противном случае — поле длины.

Если следующие за полем длины два байта выставлены в 0xFFFF, то это кадр Raw 802.3. В противном случае мы имеем либо кадр типа 802.3/LLC, либо кадр типа Ethernet SNAP, которые можно различить по значению полей SSAP и DSAP. Если они выставлены в 0xAA, то имеем кадр Ethernet SNAP, иначе — кадр типа 802.3/LLC.

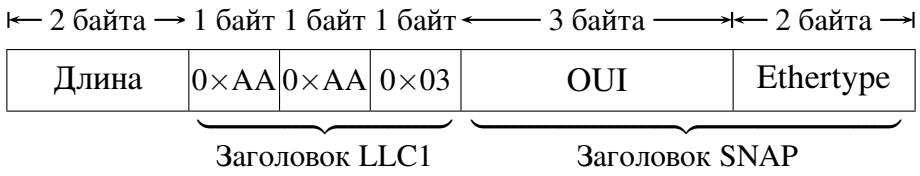


(а) Тип кадра Ethernet DIX



(б) Тип кадра Raw 802.3

(в) Тип кадра 802.3/LLC



(г) Тип кадра Ethernet SNAP

Рис. 4.8. Типы кадров Ethernet (только поле, зависящее от типа кадра)

В таблице 4.1 приведены данные об использовании разных кадров Ethernet протоколами более высоких уровней.

Таблица 4.1

Использование разных типов кадров Ethernet протоколами высших уровней

Тип кадра	Протоколы
Ethernet DIX	IPX, IP, AppleTalk Phase I
Raw 802.3	IPX
802.3/LLC	IPX, NetBEUI
Ethernet SNAP	IPX, IP, AppleTalk Phase II

4.3.3. Технология Fast Ethernet

В 1992 г. группой производителей сетевого оборудования было образовано некоммерческое объединение Fast Ethernet Alliance, целью которого стала разработка стандарта на технологию, обобщающую достижения отдельных компаний в области Ethernet-преемственного высокоскоростного стандарта. Технология получила название Fast Ethernet, и в 1995 г. комитет IEEE принял её спецификацию в качестве стандарта IEEE 802.3u.

Технология Fast Ethernet представляет собой эволюционное развитие технологии Ethernet. В данной технологии такие же формат кадра, механизм доступа к среде CSMA/CD и топология, как и в Ethernet. Эволюция коснулась нескольких элементов конфигурации средств физического уровня, включая типы применяемого кабеля, длину сегментов и количество концентраторов, что позволило увеличить пропускную способность.

Технология Fast Ethernet может использовать различные типы кабеля: витую пару разной категории, оптоволокно, причём по сравнению с Ethernet меняется как количество используемых проводников (для витой пары), так и методы кодирования. При кодировании сигнала применяются методы NRZI и MLT-3, при физическом кодировании — 4B/5B и 8B/6T.

В технологии Fast Ethernet реализована возможность выбора наиболее эффективного режима работы двух взаимодействующих портов: скорость передачи — 10 или 100 Мбит/с, вид передачи данных — дуплекс (full-duplex mode) или полудуплекс. Кроме того, во время выбора режима работы осуществляется проверка целостности линии. В режиме full-duplex вместо CSMA/CD используется соединение P2P (точка–точка) и отсутствует понятие коллизий — каждый узел может одновременно передавать и принимать кадры данных. Работа в данном режиме возможна только при соединении сетевого адаптера с коммутатором или же при непосредственном соединении коммутаторов.

4.3.3.1. Физическое соединение

Физически и логически сети на базе технологии Fast Ethernet имеют топологию звезда.

В качестве физической среды может использоваться:

- витая пара:
  - 2 пары UTP CAT.5 (100Base-TX),
  - 2 пары STP (100Base-TX),
  - 4 пары UTP CAT. 3, 4, 5 (100Base-T4);
- оптоволокно (100Base-FX).

4.3.3.2. Достоинства и недостатки технологии Fast Ethernet

Достоинства:

- увеличение пропускной способности сегментов сети до 100 Мбит/с;
- сохранение совместимости с методом случайного доступа CSMA/CD;
- сохранение формата кадра Ethernet;
- сохранение топологии звезда при построении сети;
- поддержка традиционных сред передачи данных — витой пары и оптоволоконного кабеля.

Недостатки (унаследованы от Ethernet):
— большие задержки доступа к среде при коэффициенте использования среды выше 30–40%, что связано с применением алгоритма доступа CSMA/CD;
— небольшие расстояния между узлами даже при использования оптоволокна, что связано с работой метода обнаружения коллизий;
— отсутствие механизмов выбора резервных связей;
— отсутствие поддержки приоритетного трафика приложений реального времени.

4.3.4. Технология Gigabit Ethernet

В 1995 г. группой производителей сетевого оборудования было образовано некоммерческое объединение Gigabit Ethernet Alliance, целью которого стала разработка стандарта на технологию, обобщающую достижения отдельных компаний в области Ethernet-преемственного высокоскоростного стандарта. Технология получила название Gigabit Ethernet, и в 1998 г. комитет IEEE принял её спецификацию в качестве стандарта IEEE 802.3z.

Технология Gigabit Ethernet представляет собой эволюционное развитие технологии Fast Ethernet. В данной технологии используются такой же формат кадра (за исключением длины кадра — все кадры с длиной меньше 512 байт расширяются до 512 байт), механизм доступа к среде CSMA/CD и топологию. Изменения (как и в технологии Fast Ethernet) произошли как на физическом уровне, так и на уровне MAC, в частности изменились аппаратная составляющая, физическое кодирование, параметры сети.

Технология Gigabit Ethernet может использовать в качестве среды передачи как витую пару, так и оптоволокно, причём по сравнению с Ethernet и Fast Ethernet меняется как количество используемых проводников (для витой пары), так и методы кодирования. При кодировании сигнала применяются методы NRZI и MLT-3, при физическом кодировании — 8B/10B.

В технологии Gigabit Ethernet (как и в Fast Ethernet) возможна как дуплексная (full-duplex mode), так и полудуплексная передача данных. В режиме full-duplex вместо CSMA/CD используется соединение P2P (точка–точка) и отсутствует понятие коллизий — каждый узел одновременно передаёт и принимает кадры данных. Работа в данном режиме возможна только при соединении сетевого адаптера с коммутатором или же при непосредственном соединении коммутаторов.

4.3.4.1. Физическое соединение

Физически и логически сети на базе технологии Gigabit Ethernet имеют топологию звезда.

В качестве физической среды может использоваться:
— витая пара:
  – 4 пары UTP CAT.5 (1000Base-T),
  – 2 пары STP (100Base-CX);
— мультимодовый оптоволоконный кабель с длиной волны светового сигнала 850 нм (1000Base-SX);
— мультимодовый оптоволоконный кабель с длиной волны светового сигнала 1300 нм (1000Base-LX);
— одномодовый оптический кабель (1000Base-LH).

4.3.4.2. Проблемы технологии Gigabit Ethernet и их решение

Проблемы технологии Gigabit Ethernet:

— обеспечение приемлемого диаметра сети для работы на разделяемой среде — в связи с ограничениями, накладываемыми методом CSMA/CD на длину кабеля, версия Gigabit Ethernet для разделяемой среды допускала бы длину сегмента всего 25 м;
— достижение битовой скорости 1 Гбит/с на оптическом кабеле — технология Fibre Channel, физический уровень которой был взят за основу для оптоволоконной версии Gigabit Ethernet, обеспечивает скорость передачи данных всего 800 Мбит/с;
— использование в качестве кабеля витой пары.

Для расширения максимального диаметра сети Gigabit Ethernet в полудуплексном режиме до 200 м был увеличен минимальный размер кадра — с 64 до 512 байт (без учёта преамбулы). Это повлекло за собой увеличение времени двойного оборота до 4095 bt, что сделало допустимым диаметр сети около 200 м при использовании одного повторителя.

Для увеличения длины кадра до требуемой величины сетевой адаптер должен дополнить поле данных до длины 448 байт так называемым расширением (extention), представляющим собой поле, заполненное запрещёнными символами кода 8B/10B, которые невозможно принять за коды данных.

Для сокращения накладных расходов при использовании слишком длинных кадров для передачи коротких квитанций узлам разрешено передавать несколько кадров подряд, без передачи среды другим станциям. Такой режим получил название *Burst Mode — форсированный режим передачи данных*. Станция может передать подряд несколько кадров с общей длиной не более 65 536 бит или 8192 байт. Если станции нужно передать несколько небольших кадров, то она может не дополнять их до размера в 512 байт, а передавать подряд до исчерпания предела в 8192 байт (в этот предел входят все байты кадра, в том числе преамбула, заголовок, данные и контрольная сумма).

4.3.4.3. Достоинства и недостатки технологии Gigabit Ethernet

Достоинства:

— увеличение пропускной способности сегментов сети до 1 Гбит/с;
— сохранение совместимости с методом случайного доступа CSMA/CD;
— сохранение формата кадра Ethernet;
— сохранение звездообразной топологии сетей и поддержка традиционных сред передачи данных — витой пары и оптоволоконного кабеля.

Недостатки (унаследованы от Ethernet):

— большие задержки доступа к среде при коэффициенте использования среды выше 30–40%, что связано с применением алгоритма доступа CSMA/CD;
— небольшие расстояния между узлами даже при использовании оптоволокна, что связано с работой метода обнаружения коллизий;
— отсутствие механизмов выбора резервных связей;
— отсутствие поддержки приоритетного трафика приложений реального времени.

4.4. Сети с маркерным доступом

4.4.1. Технология Token Bus

4.4.1.1. Схема передачи данных

Технология Token Bus определяет метод доступа к шине с передачей маркера. При инициализации сети станции образуют кольцо, и в соответствии с их адресами им присваиваются номера от старших к младшим. Инициализация кольца осуществляется следующим образом. В начальный момент станция включается и слушает канал. Если она не обнаруживает признаков передачи, то генерирует маркер. Если других станций не обнаружилось, то станция устанавливает кольцо из себя самой. Периодически станция генерирует специальные кадры, приглашая другие станции включиться в кольцо. Если в начальный момент сразу две станции были включены, то запускается алгоритм разрешения коллизий.

После процедуры инициализации кольца станция, имеющая наибольший номер, может послать первый кадр. Передача осуществляется в течение определённого промежутка времени, по истечении которого станция должна передать маркер следующей станции. Передача кадра разрешена только станции, владеющей маркером. Если у станции нет данных для передачи, то она передаёт маркер дальше. Коллизий в сети на базе Token Bus не возникает, так как по сети циркулирует только один маркер и только одна станция может передавать данные.

Следует отметить, что на порядок передач влияют только логические номера станций, а не их физическое размещение. Маркер передаётся только логическому соседу.

4.4.1.2. Схема приоритетов

Технология Token Bus определяет четыре приоритета для кадров: 0, 2, 4 и 6. Если маркер попадает на станцию с приоритетом 6 и у неё есть кадр на передачу, то она его передаёт. Если нет, то маркер передаётся станции с приоритетом 4. Эта подстанция передаёт свои кадры в течение своего интервала времени либо по истечении определённого промежутка передаёт маркер подстанции с приоритетом 2. Так продолжается до тех пор, пока подстанция с приоритетом 0 не перешлёт свои кадры или её таймер не исчерпается и она отдаст маркер следующей станции. Станция с наивысшим приоритетом используется для передачи трафика реального времени.

4.4.1.3. Поддержка логического кольца

Процедура поддержки логического кольца применяется при включении и выключении станций. После процедуры инициализации кольца интерфейс каждой станции хранит адреса предшествующей и последующей станций в кольце. Периодически станция, удерживающая маркер, рассылает специальный кадр, предлагая новым станциям присоединиться к кольцу. В этом кадре указаны адрес отправителя и адрес следующей за ним станции в кольце. Станции с адресами в этом диапазоне адресов могут присоединиться к кольцу. Таким образом сохраняется упорядоченность адресов в кольце. Если ни одна станция не откликнулась на посланный кадр, то станция, удерживающая маркер, закрывает окно ответа и продолжает функционировать в обычном режиме. Если есть ровно один отклик, то откликнувшаяся станция включается в кольцо и становится следующей в кольце.

Если две или более станции откликнулись, то фиксируется коллизия, и станция, удерживающая маркер, запускает алгоритм разрешения коллизий.

Если станция решила отсоединиться от сети, то после получения маркера она посылает последующей станции специальный кадр, указывающий, что её предшественником будет станция, ранее предшествующая отсоединяющейся станции. После этого происходит отсоединение станции.

4.4.1.4. Физическое соединение

Физически шина с маркером имеет линейную или древовидную топологию. Логически станции объединены в кольцо.

В качестве физической среды используется 75-омный коаксиальный кабель или витая пара. Сеть способна обеспечить пропускную способность до 10 Мбит/с при полосе пропускания кабеля 12 МГц.

4.4.1.5. Формат блока данных

В сети Token Bus циркулируют два типа блока данных: *блоки маркеров* (рис. 4.9) и *блоки данных/команд* (рис. 4.10).

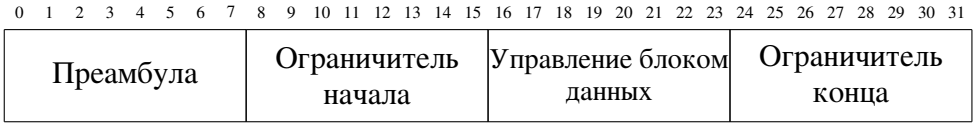


Рис. 4.9. Формат маркера Token Bus

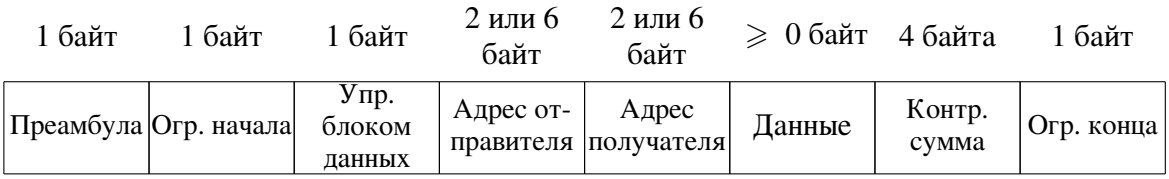


Рис. 4.10. Формат блока данных Token Bus

Поле *преамбула (Preamble)* (1 или более байт) предназначено для синхронизации таймера получателя.

Поле *ограничитель начала (Start Delimiter)* (длина 1 байт) указывает на начало маркера (или блока данных/команд), содержит сигнальные структуры, которые отличают его от остальной части блока данных.

Поле *управление блоком данных (Frame Control)* (длина 1 байт) указывает на размер адресных полей (2 или 6 байт), на тип кадра (маркер или управляющий/информационный), на приоритет кадров, может содержать подтверждения корректного или некорректного получения кадра, а также другую управляющую информацию (например, коды команд для включения станции в кольцо или исключения станции из кольца).

Поле *ограничителя конца (End Delimiter)* (длина 1 байт) содержит неинформационные символы, указывающие на конец маркера или блока данных/команд.

Поля *адрес отправителя (Source Address)* и *адрес получателя (Destination Address)* идентифицируют станции пункта назначения и источника, длина адресов может быть 6 байт или 2 байта.

Поле *данные (Data)* может иметь длину не более 8182 байт при 2-байтном адресе и 8174 байт при 6-байтном адресе, что в пять раз больше, чем в стандарте IEEE 802.3.

Поле *контрольная сумма (Frame Check Sequence)* содержит контрольную сумму, используемую для контроля ошибок. Если имеется повреждение, то блок данных отбрасывается.

4.4.1.6. Достоинства и недостатки

Достоинства:

— сеть может быть сконфигурирована для гарантированного пропускания определённого трафика, например цифрового голоса или мультимедиа;
— сеть имеет хорошую нагрузочную характеристику и неплохо работает при высоких нагрузках.

Следует отметить, что данная технология устарела и сейчас не используется, что и является её недостатком.

4.4.2. Технология Token Ring

Технология Token Ring разработана компанией IBM в 1970-х гг. Сети, построенные на базе Token Ring, были рассчитаны на скорость обмена 4 и 16 Мбит/с при числе сегментов до 250. IEEE в 1985 г. приняла данную технологию в качестве стандарта IEEE 802.5. При этом в стандарте IEEE 802.5 топология не оговорена, а сетевая среда не регламентирована.

4.4.2.1. Схема передачи данных

Станция может начать передачу данных только после получения от предыдущей станции специального кадра — маркера доступа.

Если станция готова к передаче данных, то

1) узел-отправитель:
   — ждёт получения маркера,
   — захватывает маркер (на определённое время, после истечения которого станция обязана завершить передачу своего очередного кадра и передать маркер доступа следующей станции),
   — меняет в маркере один бит, преобразующий маркер во флаг начала кадра, вносит в кадр информацию, подлежащую пересылке,
   — посылает кадр следующей станции[1];
2) переданный в сеть кадр будет двигаться по сети от станции к станции, пока не попадёт в узел, которому он адресован;
3) узел назначения:
   — копирует необходимую информацию,

[1]Когда информационный блок циркулирует по кольцу, маркер в сети отсутствует. Поэтому другие станции, желающие передать информацию, вынуждены ожидать.

— устанавливает флаг копирования (FCI), подтверждающий успешную доставку кадра адресату,

— возвращает кадр в сеть;

4) кадр продолжает движение по сети от станции к станции, пока не попадёт в узел-отправитель, где он будет уничтожен; путём контроля API (индикатора распознавания кадра адресатом) проверяется, подключена ли к сети станция назначения.

4.4.2.2. Система приоритетов

В кадре Token Ring за управление доступом отвечают два поля — *приоритет* и *резервирование*.

Станция может завладеть маркером только, если её приоритет равен или выше приоритета маркера. Если маркер уже захвачен и преобразован в информационный кадр, то только станция с приоритетом выше, чем у станции отправителя, может зарезервировать маркер на следующий цикл.

Станции, которые подняли приоритет маркера, должны его восстановить после завершения передачи.

4.4.2.3. Физическое соединение

Топологию сети Token Ring можно рассматривать с двух позиций:

— логически — кольцо,

— физически — звезда.

Отдельные станции присоединяются к сети через специальные концентраторы — *многостанционные устройства доступа (MultiStation Access Unit, MSAU)*, которые соединены между собой, образуя кольцо (рис. 4.12 и 4.11). MSAU может выполнять следующие функции: централизовывать задание конфигурации, отключать неисправные станции, контролировать работу сети и т.д. Для присоединения кабеля к MSAU применяются специальные разъёмы, которые обеспечивают замкнутость кольца даже при отключении абонента от сети. Кабель содержит в себе две разнонаправленные линии связи. В составе MSAU имеются шунтирующие реле для исключения станций из кольца.

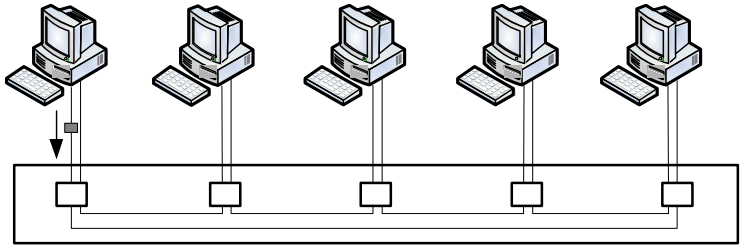


Рис. 4.11. Подсоединение узлов сети Token Ring через концентратор

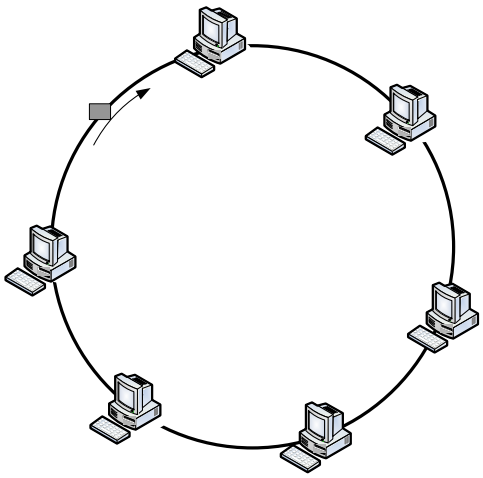


Рис. 4.12. Кольцо Token Ring

4.4.2.4. Механизмы обнаружения и предотвращения сетевых сбоев и ошибок

В сетях Token Ring существует несколько механизмов обнаружения и предотвращения сетевых сбоев и ошибок:

- присвоение одной из станций функций *активного монитора*, который играет роль центрального источника синхронизации для других станций сети, удаляет из кольца бесконечно циркулирующие кадры, генерирует новые кадры, осуществляет контроль работоспособности сети путём вывода из кольца станций, являющихся источником дефективных кадров;
- перепрограммирование MSAU для проверки наличия проблем и выборочного удаления при необходимости станций из кольца;
- применение «сигнализирующего» (*beaconing*) алгоритма:
  - станция, обнаружившая неисправность сети, высылает сигнальный блок данных, указывающий *домен неисправности*, состоящий из станции, сообщающей о неисправности, её ближайшего активного соседа, расположенного дальше по течению потока информации, и всего, что находится между ними;
  - сигнализация инициализирует *процесс автореконфигурации* (*autoreconfiguration*), в ходе которого узлы, расположенные в пределах отказавшего домена, автоматически выполняют диагностику, пытаясь реконфигурировать сеть вокруг отказавшей зоны.

4.4.2.5. Формат блока данных

В сетях на базе Token Ring циркулируют два типа блока данных: *блоки маркеров* (рис. 4.13) и *блоки данных/команд* (рис. 4.14).

Блок маркера имеет длину 3 байта. Блок данных и блок команд могут иметь разные размеры в зависимости от размеров информационного поля. Блоки данных переносят информацию для протоколов более высоких уровней, а блоки команд содержат управляющую информацию.

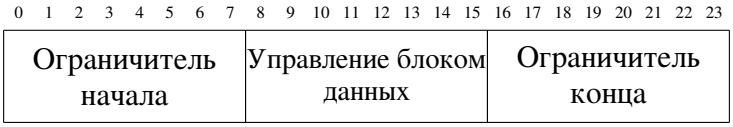


Рис. 4.13. Формат маркера Token Ring

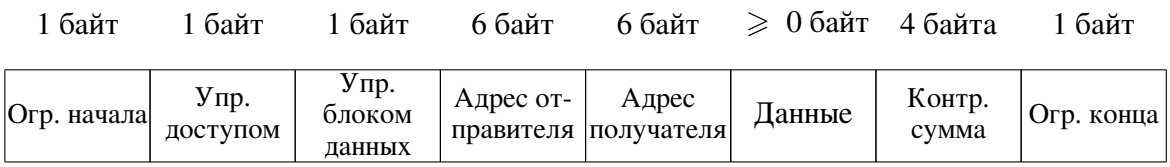


Рис. 4.14. Формат блока данных/команд Token Ring

Поле *ограничитель начала (Start Delimiter)* (длина 1 байт) указывает на начало маркера (или блока данных/команд), содержит сигнальные структуры, которые отличают его от остальной части блока данных.

Поле *управление доступом (Access Control)* (длина 1 байт) содержит следующие поля:

— поле приоритета,
— поле резервирования,
— бит маркера, используемый для дифференциации маркера и блока данных/команд,
— бит монитора, используемый активным монитором для определения, циркулирует какой-либо блок в кольце непрерывно или нет.

Поле *ограничителя конца (End Delimiter)* (длина 1 байт) сигнализирует о конце маркера (или блока данных/ команд), содержит также бит для индикации повреждённого блока и бит идентификации блока, являющегося последним в логической последовательности.

Поле *управление блоком данных (Frame Control)* (длина 1 байт) указывает на тип содержимого блока — данные или управляющая информация. В управляющих блоках это поле определяет тип управляющей информации.

Поля *адрес отправителя (Source Address)* и *адрес получателя (Destination Address)* идентифицируют станции пункта назначения и источника. Для IEEE 802.5 длина адресов равна 6 байтам.

Поле *данные (Data)* содержит передаваемые данные. Длина этого поля ограничена временем удержания маркера кольца.

Поле *контрольная сумма (Frame Check Sequence)* содержит контрольную сумму, зависящую от содержания блока данных, при помощи которой проверяется целостность кадра.

4.4.2.6. Применение

Сеть на базе технологии Token Ring может применяться для приложений, требующих предсказуемости задержки получения информации и высокой надёжности, например в сетях сопряжения с мейнфреймами.

4.4.2.7. Достоинства и недостатки

Достоинства:
- в сетях на базе технологии Token Ring не может быть коллизий, так как передавать информацию по сети может только одна станция, захватившая маркер, остальные станции вынуждены ожидать освобождения маркера;
- можно вычислить максимальное время, которое пройдёт, прежде чем любая станция сети сможет начать передачу данных.

Недостатки:
- технология Token Ring представляет собой проприетарный стандарт (IBM);
- технология Token Ring практически прекратила своё развитие;
- построение сетей на базе технологии Token Ring не получило распространения.

4.4.3. Технология FDDI

Сеть FDDI (Fiber Distributed Data Interface — волоконно-оптический распределённый интерфейс данных) представляет собой волоконно-оптическое маркерное кольцо со скоростью передачи данных 100 Мбит/с.

Стандарт FDDI был разработан комитетом X3T9.5 (впоследствии переименован в X3T12) ANSI в середине 1980-х гг. После завершения работы над FDDI ANSI представила его на рассмотрение в ISO. ISO разработала международный вариант FDDI, который полностью совместим с вариантом стандарта, разработанного ANSI.

4.4.3.1. Схема передачи данных

Двойное кольцо в сети FDDI рассматривается как общая разделяемая среда передачи данных, для которой в качестве метода доступа определён *метод маркерного кольца*, который близок к методу доступа сетей Token Ring.

Станция может начать передачу данных только после получения от предыдущей станции специального кадра — маркера доступа. Маркер — сигнал управления, состоящий из уникальной последовательности символов, которая циркулирует по кольцу после каждой информационной передачи. Если же в момент принятия маркера у станции нет данных для передачи по сети, то она немедленно передаёт маркер следующей станции.

Если станция готова к передаче данных, то
- узел-отправитель:
  - ждёт получения маркера,
  - захватывает маркер (на определённое время — *время удержания маркера (Token Holding Time, THT)*, после истечения которого станция обязана завершить передачу своего очередного кадра и передать маркер доступа следующей станции),
  - меняет в маркере один бит, преобразующий маркер во флаг начала кадра, вносит в кадр информацию, подлежащую пересылке, посылает кадр следующей станции[1];

[1]Когда информационный блок циркулирует по кольцу, маркер в сети отсутствует. Поэтому другие станции, желающие передать информацию, вынуждены ожидать.

— переданный в сеть кадр будет двигаться по сети от станции к станции, пока не попадёт в узел, которому он адресован;
— узел назначения:
  – копирует кадр в свой внутренний буфер,
  – проверяет корректность полученного кадра (в основном по контрольной сумме),
  – передаёт поле данных кадра для последующей обработки протоколу вышележащего уровня,
  – в исходном кадре отмечает следующие признаки: распознавание адреса, копирование кадра и отсутствие или наличие в нём ошибок,
  – возвращает кадр в сеть;
— вновь переданный в сеть кадр будет двигаться по сети от станции к станции, пока не попадёт в исходный узел-отправитель;
— узел-отправитель:
  – получив кадр, проверяет признаки кадра (получен ли кадр станцией назначения, был ли повреждён[1]),
  – удаляет кадр из сети,
  – передаёт маркер доступа следующей станции.

4.4.3.2. Механизм адаптивного планирования нагрузки

В сетях на базе технологии FDDI вместо системы приоритетов и резервирования, используемой в сетях на базе технологии Token Ring, применяется механизм адаптивного планирования нагрузки.

Каждая станция сравнивает реальное *время обращения маркера по кольцу (Token Rotation Time, TRT)* с заранее установленным *контрольным временем прибытия маркера (Target Token Rotation Time, TTRT)*, после чего делается вывод о слабой или сильной загруженности сети. При слабой загрузке сети станция может использовать асинхронный режим передачи информации (т.е. осуществить передачу дополнительных данных независимо от других станций). При сильной загруженности сети станция может применять только синхронный режим передачи данных, при котором передача осуществляется лишь в течение выделенного времени.

4.4.3.3. Физическое соединение

Топологию сети, построенной на базе технологии FDDI, можно рассматривать с двух позиций:
— физически:
  – двойное кольцо без деревьев,
  – двойное кольцо с деревьями,
  – дерево;

[1]Процесс восстановления информационных кадров осуществляют протоколы более высоких уровней.

— логически:

  – разделяемое кольцо.

При этом первичное кольцо используется для передачи данных, а вторичное кольцо является дублирующим 4.15.

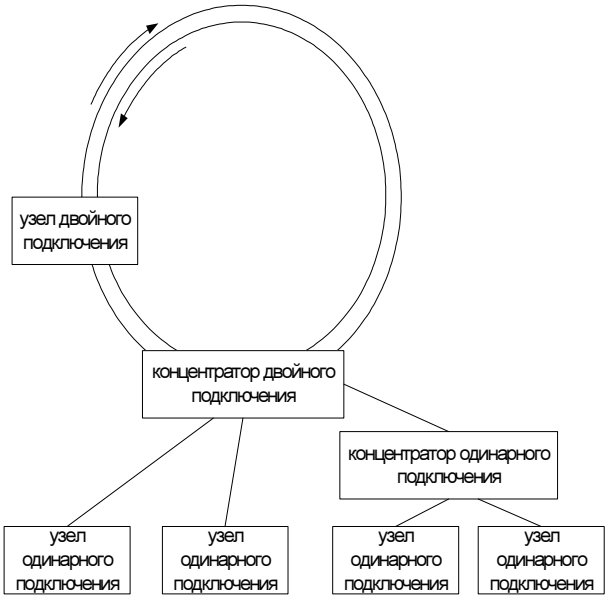


Рис. 4.15. Двойное кольцо FDDI

Физически кольцо состоит из двух или более двухточечных соединений между смежными станциями. Трафик по кольцам движется в противоположных направлениях.

Оборудование сети:

— станции:

  – *станции двойного подключения (Dual-Attachment Stations, DAS)* — подключаются как к внутреннему, так и к внешнему кольцу сети,

  – *станции одинарного подключения (Single-Attachment Stations, SAS)* — подключаются только к внешнему кольцу сети и только через концентратор или обходной коммутатор, имеющий возможность отключить их при сбое;

— *связующие концентраторы (Wiring Concentrators)* — представляют собой точки подключения к сети, выполняют также функции управления, такие как контроль работы сети, диагностика неисправностей, реконфигурация сети; бывают двух типов:

  – *концентраторы двойного подключения (Dual-Attachment Concentrator, DAC)* — подключаются как к внутреннему, так и к внешнему кольцу сети,

  – *концентраторы одинарного подключения (Single-Attachment Concentrator, SAC)* — подключаются только к внешнему кольцу сети;

— *обходные коммутаторы (Bypass Switches)* — располагаются между станцией и кольцом и позволяют отключить станцию от сети при возникновении сбоев, замкнув сигнал на себя.

4.4.3.4. Основные параметры

Основные параметры сети FDDI:

— поддержка до 500 узлов с максимальным расстоянием между соседними узлами 2 км (45 км — если используется одномодовый оптоволоконный кабель)[1];
— максимальная длина кольца — 20 км (200 км, если используется одномодовый оптоволоконный кабель, по 100 км на кольцо)[2];
— переменный размер кадра (до 4500 байт);
— длина волны — 1300 нанометров;
— максимальная скорость передачи — 100 МБод или 12.5 Мбит/с[3];
— реальная скорость работы — 80 МБод или 10 Мбит/с;
— рабочая частота — 125 МГц;
— основной вид кабеля — многомодовый или более качественный одномодовый *(Single Mode Fiber, SMF)*[4] оптоволоконный кабель,
— разъём — оптический разъём *MIC (Media Interface Connector)* (или разъём SMF-MIC для SMF-кабеля)[5], который обеспечивает подключение двух волокон кабеля, соединённых с вилкой MIC, к двум волокнам порта станции, соединённых с розеткой MIC;
— источник света — светодиоды (LED) или лазерные диоды с длиной волны 1, 3 мкм;
— метод кодирования сигнала — MLT-3;
— метод физического кодирования — 4B/5B.

4.4.3.5. Отказоустойчивость сетей на базе технологии FDDI

Основным способом обеспечения отказоустойчивости является подключение станций к двум кольцам. В нормальном режиме работы сети данные передаются по внешнему кольцу, а внутреннее кольцо при этом не используется. При возникновении сбоя в сети внешнее кольцо объединяется с внутренним, образуя таким образом единое кольцо. Данную операцию осуществляют концентраторы и/или сетевые адаптеры FDDI.

Другим способом обеспечения отказоустойчивости является использование различных процедур, определяющих наличие отказа в доступе к сети и производящих необходимую реконфигурацию. При единичном отказе сеть полностью

[1]Ограничение связано с затуханием сигнала в кабеле.

[2]Ограничение связано с необходимостью ограничения времени полного прохождения сигнала по кольцу для обеспечения предельно допустимого времени доступа.

[3]Бод — единица измерения скорости цифрового потока. Для некодированного цифрового сигнала 1 Бод=1 бит/с. Для кодирования с избыточностью — скорости разные. МБод — миллион сигналов в секунду.

[4]В этом случае дальность физического соединения между соседними узлами может увеличиться до 40–60 км в зависимости от качества кабеля, разъёмов и соединений.

[5]Кроме разъёмов MIC допускается использование разъёмов ST и SC.

восстанавливает свою работоспособность, а при множественных отказах сеть распадается на несколько несвязанных, но функционирующих сетей.

Ещё одним способом обеспечения отказоустойчивости является метод доступа к среде, т.е. использование метода маркерного кольца, который исключает возникновение коллизий и позволяет с высокой степенью вероятности просчитать время передачи маркера или данных.

4.4.3.6. Формат блока данных

В сетях FDDI циркулируют два типа блока данных: *маркеры* (рис. 4.16) и *блоки данных/команд* (рис. 4.17).

⩾ 2 байт	1 байт	1 байт	1 байт
Преамбула	Ограничитель начала	Управление блоком данных	Ограничитель конца

Рис. 4.16. Формат маркера FDDI

⩾ 2 байта	1 байт	1 байт	6 байт	байт	⩾ 0 байт	4 байта	1 байт	⩾ 2 байта
Преамбула	Огр. начала	Упр. блоком данных	Адрес отправителя	Адрес получателя	Данные	Контр. сумма	Огр. конца	Конец кадра

Рис. 4.17. Формат блока данных FDDI

Блок маркера без преамбулы имеет длину 3 байта. Блок данных и блок команд могут иметь разные размеры в зависимости от размеров информационного поля. Блоки данных переносят информацию для протоколов более высоких уровней, а блоки команд содержат управляющую информацию.

Поле *преамбула (Preamble)* (2 или более байт) используется для синхронизации. Первоначально имеет размер 8 байт, но станции, через которые проходит кадр, могут менять (уменьшать) её размер.

Поле *ограничитель начала (Start Delimiter)* (длина 1 байт) указывает на начало маркера (или блока данных/команд), содержит сигнальные структуры, которые отличают его от остальной части блока данных.

Поле *управление блоком данных (Frame Control)* (длина 1 байт) указывает на размер адресных полей (2 или 6 байт), на тип кадра (синхронный/асинхронный и управляющий/информационный), а также может содержать другую управляющую информацию (например, коды команд для управляющего кадра).

Поле *ограничителя конца (End Delimiter)* (длина 1 байт) содержит неинформационные символы, указывающие на конец маркера (или блока данных/команд).

Поля *адрес отправителя (Destination Address)* и *адрес получателя (Source Address)* идентифицируют станции пункта назначения и источника, длина адресов может быть 6 байт (по аналогии с Ethernet и Token Ring) или 2 байта. При этом поле адреса назначения может содержать индивидуальный, групповой или широковещательный адрес, в то время как адрес источника идентифицирует только одну станцию, отправившую блок данных.

Поле *данные (Data)* (0 до 4478 байт) содержит либо информацию, предназначенную для протокола высшего уровня, либо управляющую информацию.

Поле *контрольная сумма (Frame Check Sequence)* содержит контрольную сумму, зависящую от содержания блока данных, при помощи которой проверяется целостность кадра. Если повреждение имеется, то блок данных отбрасывается.

Поле *состояния блока данных (Frame Status)* позволяет станции источника определять, не появилась ли ошибка и был ли блок данных признан и скопирован принимающей станцией.

4.4.3.7. Применение

Сеть на базе технологии FDDI может применяться в качестве надёжной высокоскоростной магистрали или высокопроизводительной сети многоцелевого назначения с большим числом узлов.

4.4.3.8. Достоинства и недостатки

Достоинства:

- надёжность:
  - обеспечение избыточности благодаря двойной кольцевой конфигурации сети,
  - возможность сохранения работоспособности сети при единичных и множественных обрывах посредством сегментирования участков сети;
- отказоустойчивость:
  - возможность двойного соединения (Dual Homing) станции с сетью FDDI (два порта станции подключаются к двум разным концентраторам) позволяет активировать резервную связь при возникновении сбоев,
  - реализация так называемого «оптического обхода» обеспечивает прохождение светового сигнала по сети при сбоях в питании станции — световой сигнал обойдёт неактивную станцию через оптический переключатель (Optical Bypass Switch),
  - однократный обрыв кабеля в любом месте кольца приведёт к активации второго волоконно-оптического кольца, так как станции, расположенные по обе стороны обрыва, переконфигурируют путь циркуляции маркера и данных;
- встроенное управление:
  - каждый узел имеет объект управления, предоставляя большое число служб,
  - есть возможность SNMP управления.

Недостатки:

- высокая цена, обусловленная дорогими трансиверами, преобразующими электрический сигнал в оптический, и наоборот.

4.5. Технология 100VG-AnyLAN

Технология 100VG-AnyLAN стала результатом проекта компаний AT&T и HP по разработке альтернативной Fast Ethernet-технологии — 100Base-VG со скоростью передачи данных 100 Мбит/с, использующей в одной сети как кадры формата Ethernet, так и кадры формата Token Ring. Для стандартизации технологии в 1993 г. фирмами IBM и HP был образован комитет IEEE 802.12, а в 1995 г. технология 100VG-AnyLAN получила статус стандарта IEEE 802.12.

4.5.1. Элементы сети 100VG-AnyLAN

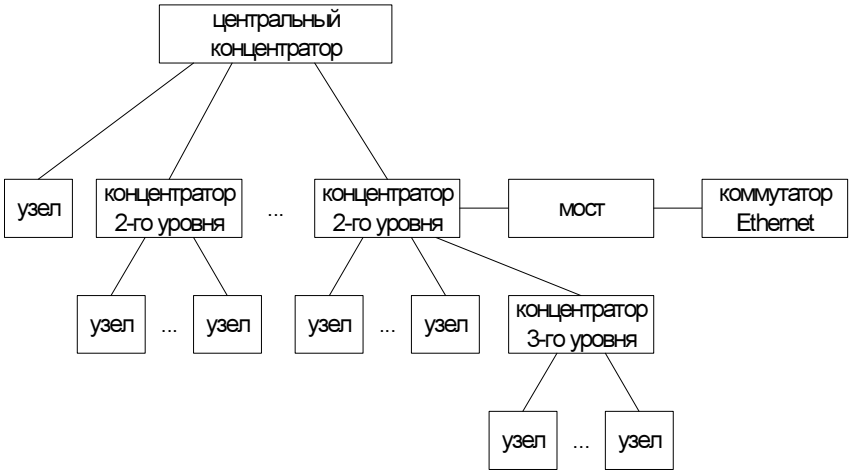


Рис. 4.18. Структура сети 100VG-AnyLAN

Сеть на базе технологии 100VG-AnyLAN (рис. 4.18) содержит следующие элементы:

- *узел* — компьютер или коммуникационное устройство технологии 100VG-AnyLAN (например, концентратор, коммутатор, мост, маршрутизатор);
- *центральный концентратор (или корневой концентратор)*:
  - управляет доступом к сети;
  - перенаправляет проходящие через него кадры узлу назначения;
- *концентратор 2-го, 3-го и т.д. уровня*:
  - имеет один порт (up-link) для присоединения в качестве узла к концентратору более высокого уровня и N портов (down-link) для присоединения узлов;
  - каждый порт может работать в одном из двух режимов: в *нормальном режиме*, когда порт передаёт только кадры, предназначенные узлу, подключённому к данному порту, или в *режиме монитора*, когда порт передаёт все кадры, обрабатываемые концентратором;
  - концентратор может быть настроен на работу или с форматом кадров Ethernet, или с форматом кадров Token Ring, причём концентраторы всей сети должны работать с пакетами только какого-нибудь одного формата;

— *мост, коммутатор или маршрутизатор* — применяются для соединения сетей 100VG-AnyLAN, использующих разные форматы кадров (кадры 802.3 Ethernet или кадры 802.5 Token Ring).

4.5.2. Схема передачи данных

Если узел сети готов к передаче данных, то

1) узел посылает концентратору, с которым он соединён, свой запрос на передачу;
2) концентратор циклически прослушивает все узлы по очереди и даёт на гарантированное время право передачи узлу, следующему по порядку за тем, который закончил передачу[1].

Узел в течение цикла кругового сканирования может передать через сеть только один кадр данных. Концентраторы, присоединённые как узлы к концентраторам верхних уровней иерархии, также выполняют свои циклы сканирования и передают запрос на передачу кадров концентратору. При этом N-портовый концентратор нижнего уровня может передать N кадров в течение цикла кругового сканирования.

4.5.3. Метод доступа Demand Priority

Метод *Demand Priority — приоритетный доступ по требованию* — представляет собой детерминированный метод разделения общей среды, использующий два уровня приоритетов: *низкий* — для обычных приложений и *высокий* — для мультимедийных приложений, чувствительных к задержкам.

Каждый концентратор имеет разные очереди для низкоприоритетных и высокоприоритетных запросов. Низкоприоритетные запросы обслуживаются до момента получения высокоприоритетного запроса.Чтобы перейти опять к обслуживанию низкоприоритетных кадров, должны быть обслужены все высокоприоритетные запросы. Чтобы обеспечить доступ для низкоприоритетных запросов в периоды высокой интенсивности поступления высокоприоритетных запросов, используется порог ожидания запроса. Если время ожидания низкоприоритетного запроса превышает этот порог, то ему присваивается более высокий приоритет.

4.5.4. Процедура подготовки к связи (Link Training)

Во время *процедуры подготовки к связи (Link Training)* концентратор и узлы обмениваются между собой управляющими пакетами специального формата. При этом проверяются правильность присоединения линий связи и их исправность, а также уровень ошибок. Одновременно концентратор получает информацию об особенностях узлов, подключённых к нему, их назначении и адресах, которые он заносит в таблицу, что позволяет ему перенаправлять полученные пакеты именно тем узлам, которым они адресованы. Концентраторы верхних уровней хранят таблицы адресов и тех узлов, которые подключены к концентраторам более низких уровней. Таким образом, основной (корневой) концентратор содержит в себе информацию о всех узлах сети.

[1]Приоритет у узлов — географический, т.е. определяется номером порта нижнего уровня, к которому подключён узел.

Запускается данная процедура подготовки к связи узлом при включении питания или после подключения к концентратору, а также автоматически при большом уровне ошибок.

4.5.5. Основные параметры сети 100VG-AnyLAN

Ниже приведены основные параметры сети на базе технологии 100VG-AnyLAN:

- топология — звезда;
- типы физической среды:
  - 4-парная неэкранированная витая пара UTP CAT 3, 4, 5,
  - 2-парная неэкранированная витая пара UTP CAT 5,
  - 2-парная экранированная витая пара STP Type 1,
  - 2-парный многомодовый или одномодовый оптоволоконный кабель;
- максимальный диаметр сети — 8 км;
- максимальная длина сегмента:
  - UTP CAT 3,4 — 100 м,
  - UTP CAT 5 — 200 м,
  - STP Type 1 — 100 м,
  - оптоволокно — 2 км,
- кодирование сигнала — NRZ (для витой пары);
- физическое кодирование — 5B/6B;
- количество уровней каскадирования концентраторов — до 5;
- максимальное количество абонентов — 1024;
- рекомендуемое количество абонентов — до 250.

4.5.6. Достоинства и недостатки

Достоинства:

- высокая скорость передачи;
- централизованный метод управления обменом без конфликтов с гарантированием предельной величины времени доступа;
- совместимость на уровне форматов кадров с сетями Ethernet или Token Ring;
- кадры передаются не всем узлам сети, а только станции назначения, что затрудняет перехват сигнала.

Недостатки:

- не обладает полной совместимостью ни с одной из стандартных сетей;
- для совместимости с сетями Ethernet или с сетями Token Ring требуется дополнительное устройство — мост.

4.6. Технологии доступа с виртуальными каналами

4.6.1. Технология X.25

Технология X.25 разработана Международным консультативным комитетом по телефонии и телеграфии в 1976 г. для организации *региональных сетей (Wide Area Network, WAN)* на базе *телефонных сетей общего пользования (ТфОП)*.

Стандарт X.25 описывает способы обмена информацией между удалёнными терминалами, локальными сетями и другими видами конечного оборудования. Стандарт предполагает обмен данными при помощи коммутации пакетов с установлением виртуальных соединений. Технология X.25 имеет свой стек протоколов с одноимённым названием, соответствующий трём нижним уровням модели ISO/OSI.

4.6.1.1. Принципы построения и компоненты сети X.25

Информационное взаимодействие в сети X.25 осуществляется на физическом, канальном и сетевом уровнях. На физическом уровне могут быть использованы любые универсальные или специализированные интерфейсы.

Компонентами сети являются устройства трёх основных категорий:

- *оконечное оборудование данных (Data Terminal Equipment, DTE)*,
- *оконечное оборудование канала передачи данных (Data Circuit-Terminating Equipment, DCE)*,
- *устройство коммутации пакетов PSE (Packet Switching Exchange)*.

Кроме того, в сети X.25 используют специальное устройство PAD (Packet Assembler/ Disassembler), предназначенное для обеспечения взаимодействия неспециализированных терминалов с сетью, для преобразования потока символов, который поступает от неспециализированного терминала в пакеты X.25, и выполнения обратного преобразования.

Для обеспечения информационного взаимодействия между компонентами сети X.25 применяется механизм организации виртуальных каналов. Между двумя терминалами устанавливается логическое виртуальное соединение на период обмена информацией, по завершении которого соединение разрывается. Существует также возможность установления постоянных виртуальных каналов.

Каждый терминал может одновременно устанавливать до 4096 виртуальных соединений или постоянных виртуальных каналов. Установленное соединение или постоянный виртуальный канал определяют маршрут движения пакетов при обмене информацией, а также скорость обмена. Скорости передачи в сетях на базе протокола X.25 определены и составляют: 1,2; 2,4; 4,8; 9,6; 19,2 Кбит/с — при использовании аналоговых абонентских линий; 64, 128, 192, 256, 384, 512, 768, 1024, 1536 и 2048 Кбит/с — при использовании выделенных линий и цифровых абонентских окончаний.

4.6.1.2. Обнаружение и коррекция ошибок

Технология X.25 разрабатывалась специально для передачи данных по линиям связи невысокого качества, в том числе по линиям ТфОП, где велика вероятность искажения информации. Поэтому основными задачами, поставленными при разработке технологии, стали:

— обеспечение передачи сообщений произвольного размера из произвольной комбинации бит;
— обеспечение выполнения процедур обнаружения ошибок на принимающей стороне;
— гарантия отсутствия дублирования и потерь компонентов (искажения) при возникновении ошибки во время передачи;
— обеспечение работы как двухточечных, так и многоточечных физических соединений;
— обеспечение подключения дуплексных и полудуплексных линий;
— обеспечение информационного обмена при значительных вариациях времени распространения сигнала.

Таким образом, одной из основных задач при разработке технологии стало обеспечение корректности принимаемых данных, для чего был использован алгоритм обнаружения и коррекции ошибок. Его принцип состоит в вычислении контрольной суммы кадра передаваемой информации и сравнении принятых данных с полученным контрольным числом на приёме. В ответ на каждый принятый кадр данных получатель должен отправить источнику подтверждение, в котором отмечено о корректности переданной информации. Источник может передавать следующую порцию данных только после получения подтверждения. Использование такого алгоритма гарантирует защиту от ошибок, возникающих при передаче, но при этом требует обмена большим количеством служебной информации, что заметно снижает скорость обмена данными между двумя терминалами.

4.6.1.3. Структура блока данных

В сетях на базе технологии X.25 в качестве протокола канального уровня используется процедура *LAPB (Link Access Procedure Balanced)*. Рекомендация X.25 определяет два основных типа процедуры LAPB — *основной* и *расширенный*, отличающихся разрядностью счётчиков, которые используются для управления потоком кадров. На рис. 4.19 приведён формат кадра LAPB.

Рис. 4.19. Формат кадра LAPB

Поле *флаг (Flag)* (1 байт) — ограничивает блок данных LAPB. Протокол LAPB использует в качестве флага комбинацию из 8 бит, которая состоит из 6 единиц и двух нулей, обрамляющих эту последовательность (01111110). Процесс приёма кадра завершается при получении следующего флага. В том случае, если к моменту получения завершающего флага приёмник получил менее 32 бит, принятый кадр считается ошибочным и уничтожается.

Поле *адрес (Address)* (1 байт) — содержит бит C/R (Command/Response), указывающий, что включает блок данных — запрос или ответ. В зависимости от значения этого бита дальше следует физический адрес принимающей или передающей станции.

Поле *управление (Control)* (1 байт) — определяет тип кадра:

— информационный кадр (Information (I) frame) — содержит информацию более высоких уровней и определённую управляющую информацию: номера последовательностей кадров и бит P/F (Poll/Final), определяющий, является ли данный кадр последним в последовательности;
— управляющий кадр (Supervisory (S) frame) — содержит управляющую информацию и не содержит информационного поля, запрашивает и приостанавливает передачу, сообщает о состоянии канала и подтверждает приём информационных кадров;
— ненумерованный кадр (Unnumbered (U) frame) — предназначен для организации и разрыва логического соединения, согласования параметров линии и формирования сигналов о возникновении неустранимых ошибок в процессе передачи данных.

Информационное поле (Data) содержит данные более высоких уровней. Если кадр не является информационным, то данное поле отсутствует.

Поле *контрольная сумма (Frame Check Sequence — FCS)* (длина 2 байта) используется для обнаружения возможных ошибок при передаче.

4.6.1.4. Применение технологии X.25

Возможные сферы применения:
— обмен сообщениями между пользователями;
— обращение большого количества пользователей к удалённой базе данных, а также к удалённому хосту электронной почты;
— связь локальных сетей (при скоростях обмена не более 512 Кбит/с);
— объединение удалённых кассовых аппаратов и банкоматов.

Иными словами, технология X.25 применяется для организации сетей, в которых трафик не является равномерным во времени, а линия связи невысокого качества.

4.6.1.5. Достоинства и недостатки сети на базе технологии X.25

Достоинства:
— в режиме реального времени есть возможность разделять один и тот же физический канал между несколькими абонентами,
— передача данных может осуществляется по каналам телефонной сети общего пользования (выделенным и коммутируемым) оптимальным образом, т.е. с максимально возможной на указанных каналах скоростью и достоверностью передачи данных,
— возможно применение механизма альтернативной маршрутизации.

Недостатки:
— невозможность передавать такие виды информации, как голос и видео.

4.6.2. Технология Frame Relay

Frame Relay (FR) — ретрансляция кадров — технология доставки сообщений в сетях передачи данных с коммутацией пакетов.

В разработке стандартов Frame Relay приняли участие три организации:

— Frame Relay Forum (FRF) — международный консорциум, включающий в себя свыше 300 поставщиков оборудования и услуг, среди которых 3Com, Northern Telecom, Digital, Cisco, Netrix, Ascom Timeplex, Newbridge Networks, Zilog и др.;
— American National Standards Institute (ANSI) — Американский национальный институт по стандартизации;
— ITU-T (International Telecommunication Union) — Международный союз электросвязи.

В 1988 г. ITU-T (в то время CCITT) принял Рекомендацию I.122 «Обеспечение дополнительного пакетного режима», которая использовалась как часть серии стандартов ISDN. Комитет ANSI T1S1 занялся развитием положений I.122, завершившимся принятием стандартов, полностью определяющих Frame Relay. Стандарт T1.606 был одобрен в 1990 г., а остальные стандарты (T1.617, T1.618) приняты в 1991 г.

4.6.2.1. Принципы построения и компоненты сети Frame Relay

Физически сети Frame Relay образуют ячеистую структуру коммутаторов. Компоненты :

— *оконечное оборудование данных (Data Terminal Equipment, DTE)*;
— *оконечное оборудование каналов передачи данных (Data Circuit-terminating Equipment, DCE)*;
— *FR-адаптеры и FR-интерфейсы (FR assembler/disassembler, FRAD)*.

Требования технологии Frame Relay:

— оконечные устройства должны поддерживать интеллектуальные протоколы более высоких уровней модели ISO/OSI;
— каналы связи должны быть свободны от ошибок;
— прикладные средства должны уметь осуществлять различные передачи.

4.6.2.2. Виртуальные каналы

Основу Frame Relay составляют *виртуальные каналы (Virtual Circuits)*. Виртуальный канал в сети Frame Relay представляет собой логическое соединение, которое создаётся между двумя устройствами DTE и используется для передачи данных.

В сети Frame Relay используется два типа виртуальных каналов — *коммутируемые (Switched Virtual Circuits, SVC)* и *постоянные (Permanent Virtual Circuits, PVC)*.

SVC устанавливается динамически. Для него стандарты передачи сигналов определяют, как узел должен устанавливать, поддерживать и сбрасывать соединение. Процесс передачи данных с использованием SVC состоит из четырёх последовательных фаз:

— *установление вызова (Call Setup)* — создаётся виртуальное соединение между двумя DTE;
— *передача данных (Data Transfer)* — фаза непосредственной передачи данных;
— *ожидание (Idle)* — виртуальное соединение ещё существует, но передача данных через него уже не производится; если период ожидания превысит установленное значение тайм-аута, соединение может быть завершено автоматически;

— *завершение вызова (Call Termination)* — фаза завершения соединения.

PVC включает в себя конечные станции, среду передачи и все коммутаторы, расположенные между конечными станциями. После установки PVC для него резервируется определённая часть полосы пропускания, и двум конечным станциям не требуется устанавливать или сбрасывать соединение.

Процесс передачи данных по каналу PVC имеет всего две фазы:

— передача данных — фаза непосредственной передачи данных;
— ожидание — виртуальное соединение существует, однако передача данных через него не производится.

В отличие от SVC, постоянный канал PVC не может быть автоматически разорван в том случае, если он не используется для передачи данных.

PVC имеют два преимущества над SVC:

— могут обеспечить более высокую производительность, так как соединение устанавливается предварительно и впоследствии не разрывается;
— обеспечивают лучший контроль над сетью, так как провайдер или сетевой администратор может выбирать путь, по которому будут передаваться кадры.

Однако и SVC имеют ряд преимуществ над PVC:

— могут имитировать сети без установления соединений (необходимо, если пользователь использует приложение, которое не может работать в сети с установлением соединения);
— используют полосу пропускания только тогда, когда это необходимо (PVC должны постоянно её резервировать на тот случай, если она понадобится);
— требуют меньшей административной работы, поскольку устанавливаются автоматически, а не вручную.

Однако режим SVC не получил широкого распространения, в силу сложности в реализации. Как следствие, PVC является наиболее распространённым режимом связи в сети FR.

4.6.2.3. Формат блока данных

На рис. 4.20 приведён формат кадра Frame Relay.

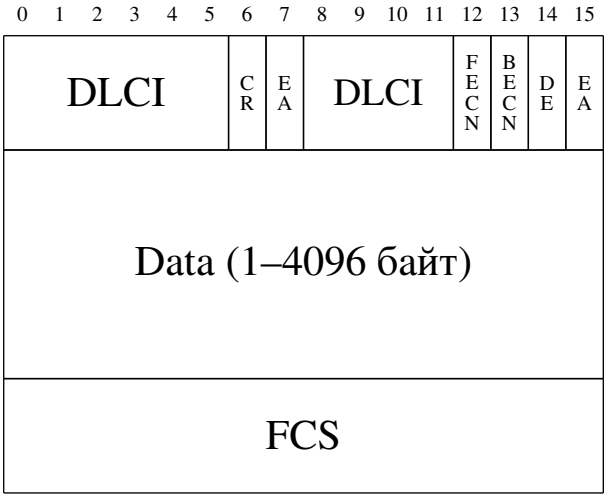


Рис. 4.20. Формат кадра Frame Relay

Поле *Флаг* обрамляет кадр Frame Relay.

Поле *Заголовок* содержит:

- поле *Идентификатор канала передачи данных (Data Link Connection Identifier, DLCI)* — определяет абонентский адрес в сети Frame Relay (стандарт FRF), состоит из шести бит первого октета и четырёх бит второго октета заголовка кадра (стандарты ANSI и ITU-T допускают размер заголовка до 4 байт);
- поле *Запрос/Ответ (Command/ Response, CR)* (2 бита) — зарезервировано для возможного применения в различных протоколах более высоких уровней модели ISO/OSI;
- бит *Расширение адреса (Extended Address, EA)* — устанавливается в конце каждого октета заголовка и указывает на наличие/отсутствие расширения заголовка Frame Relay на целое число дополнительных октетов с целью указания адреса, состоящего более чем из 10 бит, причём если бит имеет значение 1, то данный октет в заголовке последний;
- бит *Уведомление приёмника о явной перегрузке (Forward Explicit Congestion Notification, FECN)* — устанавливается аппаратурой канала данных в 1 для уведомления получателя сообщения о том, что произошла перегрузка в направлении передачи данного кадра;
- бит *Уведомление источника о явной перегрузке (Backward Explicit Congestion Notification, BECN)* — устанавливается аппаратурой канала данных в 1 для уведомления источника сообщения о том, что произошла перегрузка в направлении, обратном направлению передачи содержащего этот бит кадра, после чего источник должен снизить интенсивность передаваемого потока данных;
- бит *Разрешения сброса (Discard Eligibility, DE)* — устанавливается в 1 (либо аппаратурой канала данных, либо оконечным оборудованием) в случае явной перегрузки и указывает на то, что данный кадр может быть уничтожен в первую очередь.

Информационное поле (Data) содержит данные пользователя и состоит из целого числа октетов. Его максимальный размер определён стандартом FRF и составляет 1600 байт (минимальный размер — 1 байт), но возможны и другие максимальные размеры (вплоть до 4096 байт). Содержание информационного поля пользователя передаётся без внесения изменений.

Поле *контрольная сумма (Frame Check Sequence, FCS)* (длина 2 байта) используется для обнаружения возможных ошибок при передаче. Содержит 16-разрядную контрольную сумму всех полей кадра Frame Relay, за исключением поля *Флаг*.

4.6.2.4. Адресация в сетях Frame Relay

Для идентификации виртуальных каналов в сети Frame Relay используется DLCI, который определяет номер виртуального порта для процесса пользователя. Обычно идентификатор DLCI имеет только локальное значение и не является уникальным в пределах сети. Конкретные значения DLCI для каждого пользователя определяются провайдером сервиса Frame Relay.

Дополнение в виде глобальной адресации позволяет применять идентификаторы узлов. При использовании этого дополнения значения, вставленные в поле DLCI блока данных, являются глобально значимыми адресами индивидуальных

устройств конечного пользователя (например, маршрутизаторов). Аппаратура канала данных обязана обладать способностью определения принадлежности проходящего кадра конкретному PVC. Внутри сети Frame Relay могут использоваться различные сетевые адреса. Для разных интерфейсов одно и то же значение DLCI может применяться многократно.

4.6.2.5. Отличия протокола Frame Relay от HDLC

Отличия протокола Frame Relay от HDLC состоят в следующем:
- Frame Relay не предусматривает передачу управляющих сообщений;
- для передачи служебной информации используется специально выделенный канал сигнализации;
- отсутствует нумерация последовательно передаваемых (принимаемых) кадров, так как протокол Frame Relay не имеет никаких механизмов для подтверждения правильно принятых кадров.

4.6.2.6. Применение технологии Frame Relay

Данная технология применяется как для управления пульсирующим трафиком между локальными сетями и территориальной сетью, так и для передачи голоса.

4.6.2.7. Достоинства и недостатки

Достоинства:
- малое время задержки;
- простой формат кадров, содержащих минимум управляющей информации, следствием чего является высокая эффективность передачи данных (в предположении, что канал надёжен);
- независимость от протоколов верхних уровней модели ISO/OSI;
- предсказуемая пропускная способность;
- возможность контроля работоспособности (нагруженности) канала;
- возможность приоритезации разнородного трафика (для каждого типа трафика можно организовать своё виртуальное соединение).

Недостатки:
- Frame Relay не различает протоколы вышележащих уровней и, следовательно, нельзя приоритезировать трафик без организации дополнительных виртуальных соединений, что несёт дополнительные накладные расходы;
- отсутствие широковещательного множественного доступа;
- нет встроенных функций контроля доставки и управления потоком кадров (функции управления потоком выполняются протоколами верхних уровней).

4.7. Технологии региональных сетей

Региональные сети строятся по принципу функционального разделения по уровням доступа: опорная сеть (магистраль), уровень распределения/агрегации, уровень доступа (клиентский доступ).

4.7.1. Технологии опорной сети

Опорная сеть обычно имеет кольцевую топологию, обеспечивающую резервирование и повышенную надёжность. В качестве физической среды передачи данных применяется оптоволокно. Базовыми магистральными технологиями являются SONET/SDH, ATM, POS (Pocket over Sonet), EoSDH (Ethernet over SDH), DWDM, CWDM, DPT/RPR, Fast/Gigabit/10 Gigabit Ethernet.

На уровне доступа применяются следующие технологии: Ethernet (Ethernet, Fast Ethernet, Gigabit Ethernet), LRE, xDSL (HDSL, ADSL, VDSL, SDSL), PNA (Phoneline Networking Alliance), Wireless (802.11), Infrared, PON (Passive Optical Network), EFM (Ethernet in the First Mile, IEEE IEEE 802!802.3!802.3ah), Satellite.

Для обеспечения повышенной надёжности и резервирования широко используется топологическая модель кольца. Кольца обычно создают на уровнях опорной сети и доступа.

4.7.1.1. SONET/SDH

Изначально основной задачей телекоммуникационных структур являлась передача голосового трафика. Скорость передачи данных задаётся относительно звука с *импульсной модуляцией (Pulse Code Modulation, PCM)* с частотой дискретизации 8 кГц и 8-битной дискретизацией. В результате получается *базовый поток (Digital Signal, DS0)* 64 Кбит/с. Потоки агрегируются и передаются по высокоскоростным каналам. Агрегирование происходит по *технологии временного мультиплексирования каналов (Time Division Multiplexing, TDM)*. Непосредственное слияние и разделение каналов производят специальные устройства — *мультиплексоры*. Например, на вход мультиплексора может поступать 30 потоков DS0 (64 Кбит/с $\times$ 30 + 2 сигнальных по 64 Кбит/с), а на выходе получается один E1 (2048 Кбит/с).

В свою очередь, для мультиплексирования потоков информации при формировании мощных региональных и межрегиональных каналов были разработаны стандарты для высокоскоростных оптических сетей связи — сначала *плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH)*, а затем и более совершенная *синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH)*, распространённая в Европе, и её американский аналог *SONET*.

SONET/SDH предполагает использование метода временно́го мультиплексирования и синхронизацию временны́х интервалов трафика между элементами сети и определяет уровни скоростей прохождения данных и физические параметры. Основными устройствами являются мультиплексоры, а физической средой передачи — оптоволокно. При построении сети SDH обычно используется топология двойного кольца. По одному кольцу передаётся синхронизирующая информация, а по другому — непосредственно трафик. Использование колец даёт возможность автоматического восстановления при авариях. Метод передачи — коммутирование каналов.

К достоинствам SONET/SDH относят:
- стандартизованность,
- масштабируемость,
- высокую надёжность (время восстановления порядка 50 мс).

К недостаткам SONET/SDH относят:
- ориентацию на передачу голосового трафика,
- фиксированную полосу пропускания, не зависящую от загрузки каналов,

— неэффективное использование колец.

SONET/SDH является самой зрелой и поэтому самой распространённой на данный момент технологией для построения магистральных каналов передачи данных. Основная область её применения — первичные сети операторов связи. Мультиплексоры, объединённые оптическими линиями связи, образуют единую среду, в которой прокладываются цифровые каналы между оборудованием телефонных сетей или сетей передачи данных. Кроме того, технология SONET/SDH может являться транспортной основой для более современных протоколов, таких как ATM, POS и MPLS.

4.7.1.2. ATM

Как решение проблемы создания мультисервисной и высокоскоростной технологии передачи данных была предложена *технология асинхронной передачи данных (Asynchronous Transfer Mode, ATM)*. В локальных сетях ATM распространения не получила, но до сих пор применяется при построении магистральных сетей. ATM может работать поверх SONET/SDH.

Технология ATM представляет собой транспортный механизм коммутации ячеек небольшого размера фиксированной длины (53 байта). Наиболее распространённая среда передачи для ATM — оптоволокно.

В ATM при соединении создаётся виртуальный канал. Далее коммутация ячеек происходит на основе идентификаторов виртуального канала (VPI/VCI), присутствующих в заголовках.

ATM имеет встроенную поддержку обеспечения гарантированного качества обслуживания.

4.7.1.3. POS

Для решения проблемы накладных расходов в случае передачи IP-трафика через сети SONET/SDH с использованием ATM была разработана технология *POS (Packet Over Sonet/SDH)*, непосредственно инкапсулирующая данные в кадры SDH. Практически получается интерфейс с IP-адресом, который использует все преимущества транспортной оптической технологии, не задействуя никаких промежуточных протоколов.

4.7.1.4. EoSDH

Отвечая потребностям рынка по передаче непосредственно Ethernet трафика по наследованным оптическим сетям, появилась технология Ethernet over SONET/SDH. Вначале допускались только соединения типа точка-точка, затем возникли и многоточечные каналы.

4.7.1.5. WDM

Непрерывно возрастающие объёмы трафика требуют повышения пропускной способности оптических магистралей. Кроме тривиального повышения скоростей передачи существует и другой способ решения данной задачи — уплотнение (мультиплексирование) каналов. Наиболее развитой в настоящее время является

технология оптического спектрального уплотнения, называемая обычно *мультиплексированием с разделением по длине волны (Wavelength Division Multiplexing, WDM)*.

Принцип работы WDM следующий. Потоки данных от отдельных источников переносятся световыми волнами разной длины (каждому каналу принадлежит своя длина) и объединяются мультиплексором в единый многочастотный сигнал, который передаётся по оптическому волокну. На стороне приёмника происходит обратное преобразование.

Технология WDM соответствует физическому уровню сетевых взаимодействий и работает независимо от типа и формата передаваемых данных, то есть является протокольно-независимой. К WDM мультиплексору можно подключить практически любое оборудование: SONET/SDH, ATM, Ethernet.

WDM бывает двух видов: *плотное волновое мультиплексирование (Dense Wavelength Division, DWDM)* и *грубое волновое мультиплексирование (Coarse Wavelength Division, CWDM)*.

DWDM может обеспечить большое число спектральных каналов на одно оптоволокно (32, 64 или даже 128). Отсюда её основная отличительная особенность — малые расстояния между мультиплексными каналами.

CWDM-системы рассчитаны на меньшее число каналов (4, 8 или 16). Поэтому в них спектры соседних информационных каналов расположены на гораздо больших расстояниях друг от друга, чем в DWDM. Скорости передачи CWDM систем ниже, чем у DWDM.

4.7.1.6. DPT/RPR

Стандарт IEEE 802.17 (вобравший в себя DPT/RPR) позиционируется как высокоскоростная технология динамической передачи IP-пакетов, предназначенная для решения задач построения региональных сетей.

В DPT/RPR (IEEE 802.17) к IP-пакету добавляется прослойка второго уровня (MAC), пакет помещается в произвольную физическую оптическую среду (SONET/SDH, WDM) с топологией двойного кольца. Данные одновременно передаются по двум кольцам в противоположных направлениях. Поток данных в каждом кольце включает непосредственно транспортируемые в данном кольце данные и управляющие пакеты для соседнего кольца.

Достоинства:

— пакетно-ориентирован;
— не требуется дополнительная прослойка типа ATM для доступа к физической оптической среде;
— заложен высокий уровень резервирования и быстрая восстановимость в случае аварий (50 мс);
— эффективно используется ёмкость оптических каналов за счёт смешения контрольных и передаваемых данных.

4.7.2. Технологии уровня доступа

Существует широкий спектр решений для обеспечения абонентского доступа (так называемая «первая/последняя миля»): Ethernet (Ethernet, Fast Ethernet, Gigabit Ethernet), LRE, xDSL (HDSL, ADSL, VDSL, SDSL), PNA (Phoneline Networking Alliance), Wireless (802.11), Infrared, PON (Passive Optical Network), EFM (Ethernet in the First Mile alliance IEEE 802!802.3!802.3ah), Satellite.

4.7.2.1. VLAN

Для построения развитых Ethernet сетей используют *технологию виртуальных локальных сетей (Virtual Private Lan, VLAN)* (IEEE 802.1Q), которая позволяет создавать в едином Ethernet-сегменте независимые логические области, ограничивающие на канальном уровне распространение трафика (в том числе и широковещательного). В заголовок Ethernet-фрейма вводится дополнительная информация о принадлежности к влану (VLAN); получается помеченный кадр данных (Tagged Vlan), который передаётся по транковым линиям (802.1Q Trunk). Это позволяет передавать по одному каналу данные нескольких изолированных локальных сетей. Дальнейшая коммутация происходит с учётом 802.1Q-метки. На выходе из коммутатора (например, на стороне клиентского порта) метка (Tag) убирается, что называется *вхождением порта в нетагированный влан (Untagged Vlan)*.

Обычно клиентские подсети изолируются друг от друга путём подключения к раздельным вланам (через порты с Untagged Vlan), а связь между ними организуется при помощи маршрутизатора через 802.1Q транки.

На практике использование вланов даёт возможность гибко изменять логическую организацию сети независимо от реальной физической топологии.

4.7.2.2. Q-in-Q

Непосредственным решением присущих 802.1Q вланам ограничений (например, их максимальное число 4096) явилась технология Q-in-Q. Операторское устройство, получающее клиентский кадр Ethernet, добавляет ещё одну 802.1Q-метку, которая и принимается во внимание при дальнейшей коммутации. Так получается целый блок меток, а сам процесс называется *стекированием вланов (802.1Q stacking)*. На выходе из провайдерской сети дополнительная метка удаляется. Это позволяет строить полностью прозрачные на канальном уровне сети.

4.7.2.3. STP

В сетях Ethernet коммутаторы поддерживают только древовидные связи (ациклический граф). Отказоустойчивость требует наличия петель (циклический граф). Технология STP (Spanning Tree Protocol) позволяет совместить оба требования.

После активирования коммутаторы обмениваются специальными информационными пакетами (BPDU), с помощью которых вначале выбирается корневой мост (который будет в итоге находиться на вершине древовидной структуры), а затем кратчайшие (в смысле пропускной способности) пути от каждого из коммутаторов до корневого. В конечном итоге формируется логическая беспетельная топология путём блокирования некоторых избыточных связей.

Расширением STP является стандарт *RSTP (Rapid Spanning Tree Protocol)*.

4.7.3. Технология Metro Ethernet

Преимущества Ethernet: высокая скорость, лёгкость масштабирования технологии, простота для массового использования.

Первоначально Ethernet строилась на базе разделяемой среды передачи, но позднее был введён коммутируемый Ethernet. Были созданы механизмы, гарантирующие качество обслуживания, что дало возможность использовать Ethernet для передачи мультимедийных данных.

Развитием технологии Ethernet для региональных сетей занимается *Metro Ethernet Forum (MEF)* — некоммерческая организация, созданная для продвижения концепции построения операторских сетей на основе Ethernet и ускорения их развёртывания во всём мире. В октябре 2003 г. форум Metro Ethernet ратифицировал первый стандарт, описывающий службы Metro Ethernet: MEF Technical Specification—Ethernet Services Model Phase 1.

По сравнению с технологиями, имеющими схожие потребительские свойства, например SDH/SONET, реализация Metro Ethernet обходится в среднем в 2–3 раза дешевле. В настоящее время все серьёзные поставщики оборудования выпускают оборудование для Metro Ethernet и ведут активную маркетинговую политику по его продвижению на рынке.

Форум Metro Ethernet предложил модель услуг Metro Ethernet. В основе базовой модели лежит *городская Ethernet-сеть (Metro Ethernet Network, MEN)*, принадлежащая провайдеру. *Клиентское оборудование (Customer Equipment, CE)* подключается к сети с помощью интерфейса UNI (User Network Interface), который представляет собой стандартный Ethernet.

Для потребителя существует только Ethernet-интерфейс (UNI), которым он подключается к провайдеру услуг. Транспортные технологии, обеспечивающие работу Metro Network, для него скрыты.

Ключевым элементом модели является *виртуальное соединение Ethernet (Ethernet Virtual Connection, EVC)*, которе определяется как соединение двух и более UNI. По ним проходят данные в виде кадров Ethernet. EVC выполняет две функции:

- соединяет UNI потребителей и пропускает между ними Ethernet-фреймы; обеспечивает защищённость и безопасность;
- доставка кадров Ethernet производится с неизменяемыми параметрами: MAC-адреса и содержимое не изменяются в отличие от маршрутизирующих сетей.

MEF определяет два типа EVC: *один-к-одному (Point-to-Point)* и *многие-ко-многим (Multipoint-to-Multipoint)*.

MEF определяет два типа базовых услуг Ethernet: *E-Line (Ethernet Line service type)* и *E-LAN (Ethernet LAN service type)*:

- E-Line обеспечивает соединения point-to-point (аналог физических выделенных каналов или виртуальных выделенных каналов Frame Relay);
- E-LAN поддерживает multipoint соединения (подобен услуге прозрачных локальных сетей (TLS)).

Для полного определения сервисов провайдер услуг должен обозначить кроме типа сервиса (E-Line или E-LAN) на основе EVC ещё и атрибуты, которые можно сгруппировать по категориям:

- *физический интерфейс (Ethernet Physical Interface)* определяет параметры физического уровня модели OSI;
- *параметры трафика (Traffic Parameters)* определяют полосу пропускания;
- *дополнительные параметры качества трафика (Perfomance Parameters)*: доступность (Availability), задержка (Delay), джиттер (Jitter), потери (Loss);
- *классы обслуживания (Class of Service)*;
- *необходимость доставки служебных пакетов (Service Frame Delivery)*;
- *поддержка VLAN (Vlan Tag Support)*: 802.1q, Q-in-Q, MAC-in-MAC;
- *фильтры (Security Filters)*: разнообразная фильтрация фреймов на основе различных критериев;
- *мультиплексирование виртуальных соединений (Service multiplexing)*: поддержка нескольких EVC на одном UNI;

— *неизменность клиентских VLAN (Vlan Transparency)*: неизменность клиентских вланов CE-Vlan при переходе через UNI, т.е. входной CE-Vlan и выходной CE-Vlan для одного и того же EVC одни и те же;
— *связывание (Bundling)*: отображение нескольких вланов CE-Vlan на одно EVC (используя Q-in-Q).

4.8. Технологии беспроводного доступа

4.8.1. Методы доступа к среде в беспроводных сетях

Существует несколько базовых методов доступа (их ещё называют методами уплотнения или мультиплексирования), основанных на разделении между станциями таких параметров, как пространство, время, частота и код. Задача уплотнения — выделить каждому каналу связи пространство, время, частоту и/или код с минимумом взаимных помех и максимальным использованием характеристик передающей среды.

4.8.1.1. Уплотнение с пространственным разделением

Основано на разделении сигналов в пространстве, когда передатчик посылает сигнал, используя код c, время t и частоту f области s_i. То есть каждое беспроводное устройство может вести передачу данных только в границах определённой территории, на которой любому другому устройству запрещено передавать свои сообщения.

4.8.1.2. Уплотнение с частотным разделением (Frequency Division Multiplexing, FDM)

Каждое устройство работает на определённой частоте, благодаря чему несколько устройств могут вести передачу данных на одной территории. Это один из наиболее известных методов, так или иначе используемый в самых современных системах беспроводной связи.

Эта схема приводит к неоправданному расточительству частотных ресурсов, поскольку требует выделения своей частоты для каждого беспроводного устройства.

4.8.1.3. Уплотнение с временны́м разделением (Time Division Multiplexing, TDM)

В данной схеме распределение каналов идёт по времени, т. е. каждый передатчик транслирует сигнал на одной и той же частоте, но в различные промежутки времени (как правило, циклически повторяющиеся) при строгих требованиях к синхронизации процесса передачи.

Временны́е интервалы могут динамично перераспределяться между устройствами сети. Устройствам с большим трафиком назначаются более длительные интервалы, чем устройствам с меньшим объёмом трафика.

Основной недостаток систем с временны́м уплотнением — мгновенная потеря информации при срыве синхронизации в канале, например из-за сильных помех, случайных или преднамеренных.

4.8.1.4. Уплотнение с кодовым разделением (Code Division Multiplexing, CDM)

В данной схеме все передатчики транслируют сигналы на одной и той же частоте.

В схеме CDM каждый передатчик заменяет каждый бит исходного потока данных на CDM-символ — кодовую последовательность длиной в 11, 16, 32, 64 и т. п. бит (так называемый чип). Кодовая последовательность уникальна для каждого передатчика.

Приёмник знает CDM-код передатчика, сигналы которого должен воспринимать. Он постоянно принимает все сигналы и оцифровывает их. Затем в специальном устройстве (корреляторе) производится операция свёртки (умножения с накоплением) входного оцифрованного сигнала с известным ему CDM-кодом и его инверсией. В несколько упрощённом виде это выглядит как операция скалярного произведения вектора входного сигнала и вектора с CDM-кодом. Если сигнал на выходе коррелятора превышает некий установленный пороговый уровень, приёмник считает, что принял 1 или 0. Для увеличения вероятности приёма передатчик может повторять посылку каждого бита несколько раз. При этом сигналы других передатчиков с другими CDM-кодами приёмник воспринимает как аддитивный шум. Благодаря большой избыточности мощность принимаемого сигнала может быть сопоставима с интегральной мощностью шума. Сходства CDM-сигналов со случайным (гауссовым) шумом добиваются, используя CDM-коды, порождённые генератором псевдослучайных последовательностей. Этот метод также называется *методом расширения спектра сигнала посредством прямой последовательности (Direct Sequence Spread Spectrum, DSSS)*.

Наиболее сильная сторона данного уплотнения заключается в повышенной защищённости и скрытности передачи данных: не зная кода, невозможно получить сигнал, а в ряде случаев и обнаружить его присутствие. Кроме того, кодовое пространство более значительно по сравнению с частотной схемой уплотнения, что позволяет без особых проблем присваивать каждому передатчику свой индивидуальный код.

4.8.1.5. Механизм мультиплексирования посредством ортогональных несущих частот (Orthogonal Frequency Division Multiplexing, OFDM)

Весь доступный частотный диапазон разбивается на достаточно много поднесущих (от нескольких сот до тысяч). Одному каналу связи (приёмнику и передатчику) назначают для передачи несколько таких несущих, выбранных из множества по определённому закону. Передача ведётся одновременно по всем поднесущим, т.е. в каждом передатчике исходящий поток данных разбивается на N субпотоков, где N — число поднесущих, назначенных данному передатчику. Распределение поднесущих в ходе работы может динамически изменяться.

Преимущества:

— Селективному замиранию будут подвержены только некоторые подканалы, а не весь сигнал. Если поток данных защищён кодом прямого исправления ошибок, то с этим замиранием легко бороться.

— OFDM позволяет подавить межсимвольную интерференцию. Межсимвольная интерференция оказывает значительное влияние при высоких скоростях передачи данных, так как расстояние между битами (или символами) мало. В схеме OFDM скорость передачи данных уменьшается в N раз, что

позволяет увеличить время передачи символа в N раз. Таким образом, если время передачи символа для исходного потока составляет T_s, то период сигнала OFDM будет равен NT_s. Это позволяет существенно снизить влияние межсимвольных помех. При проектировании системы N выбирается таким образом, чтобы величина NT_s значительно превышала среднеквадратичный разброс задержек канала.

4.8.2. Стек протоколов IEEE 802.11 (WiFi)

Стандарты IEEE 802.11 (WiFi, Wi-Fi, Wireless Fidelity)[1] описывают *беспроводную технологию локальных сетей (Wireless Local Area Network, WLAN)*.

Сети WLAN имеют ряд преимуществ перед обычными кабельными сетями:
- их можно быстро развернуть;
- пользователи мобильных устройств при подключении к локальным беспроводным сетям могут легко перемещаться в рамках действующих зон сети;
- скорость современных сетей довольно высока (до 108 Мбит/с), что позволяет использовать их для решения очень широкого спектра задач;
- может оказаться единственным выходом, если невозможна прокладка кабеля для обычной сети.

Вместе с тем беспроводные сети имеют ряд ограничений:
- меньшая, чем в проводных сетях, скорость;
- подверженность влиянию помех;
- более сложная схема обеспечения безопасности передаваемой информации.

На физическом уровне существует несколько вариантов спецификаций, которые отличаются используемым частотным диапазоном, методом кодирования и как следствие — скоростью передачи данных. Все варианты физического уровня работают с одним и тем же алгоритмом уровня MAC, но некоторые временные параметры уровня MAC зависят от используемого физического уровня.

4.8.2.1. Уровень доступа к среде стандарта 802.11

В сетях 802.11 уровень MAC обеспечивает два режима доступа к разделяемой среде:
- *распределённый режим (Distributed Coordination Function, DCF)*;
- *централизованный режим (Point Coordination Function, PCF)*.

4.8.2.1.1. Распределённый режим доступа DCF. В этом режиме реализуется *метод множественного доступа с контролем несущей и предотвращением коллизий (Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA)*. Вместо неэффективного в беспроводных сетях прямого распознавания коллизий по методу CSMA/CD здесь используется их косвенное выявление. Для этого каждый переданный кадр должен подтверждаться кадром положительной квитанции, посылаемым станцией назначения. Если по истечении оговорённого времени квитанция не поступает, станция-отправитель считает, что произошла коллизия.

Режим доступа DCF требует синхронизации станций. В спецификации 802.11 временны́е интервалы начинают отсчитываться от момента окончания передачи очередного кадра. Это не требует передачи каких-либо специальных синхронизирующих сигналов и не ограничивает величину пакета размером слота, так как

[1] По аналогии с Hi-Fi.

слоты учитываются только при принятии решения о начале передачи кадра. Станция, которая хочет передать кадр, обязана предварительно прослушать среду.

Предусматривается два механизма обнаружения несущей: *физический* и *виртуальный*. Первый механизм реализован на физическом уровне и сводится к определению уровня сигнала в антенне и сравнению его с пороговой величиной. Виртуальный механизм обнаружения несущей основан на том, что в передаваемых кадрах данных, а также в управляющих кадрах АСК и RTS/CTS содержится информация о времени, необходимом для передачи пакета (или группы пакетов) и получения подтверждения. Все устройства сети получают информацию о текущей передаче и могут определить, сколько времени канал будет занят, т.е. устройство при установлении связи сообщает всем, на какое время оно резервирует канал. Как только станция фиксирует окончание передачи кадра, она обязана отсчитать интервал времени, равный межкадровому интервалу (IFS). Если после истечения IFS среда все ещё свободна, начинается отсчёт слотов фиксированной длительности. Кадр можно передавать только в начале какого-либо из слотов при условии, что среда свободна. Станция выбирает для передачи слот на основании усечённого экспоненциального двоичного алгоритма отсрочки, аналогичного используемому в методе CSMA/CD. Номер слота выбирается как случайное целое число, равномерно распределённое в интервале [0, CW], где CW (Contention Window) — *конкурентное окно*.

Если же в начале какого-нибудь слота среда оказывается занятой, то вычитания единицы не происходит, и таймер «замораживается». В этом случае станция начинает новый цикл доступа к среде, изменяя только алгоритм выбора слота для передачи. Как и в предыдущем цикле, станция следит за средой и при её освобождении делает паузу в течение межкадрового интервала. Если среда осталась свободной, то станция использует значение «замороженного» таймера в качестве номера слота и выполняет описанную выше процедуру проверки свободных слотов с вычитанием единиц, начиная с замороженного значения таймера отсрочки.

Размер слота зависит от способа кодирования сигнала (например, для метода FHSS размер слота равен 28 мкс; для метода DSSS — 1 мкс). Размер слота выбирается таким образом, чтобы он превосходил время распространения сигнала между любыми двумя станциями сети плюс время, затрачиваемое станцией на распознавание занятости среды. Если такое условие соблюдается, то каждая станция сети сумеет правильно распознать начало передачи кадра при прослушивании слотов, предшествующих выбранному ею для передачи слоту. Это, в свою очередь, означает, что коллизия может иметь место только в том случае, когда несколько станций выбирают один и тот же слот для передачи. В этом случае кадры искажаются, и квитанции от станций назначения не приходят. Не получив в течение определённого времени квитанцию, отправители фиксируют факт коллизии и пытаются передать свои кадры снова. При каждой повторной неудачной попытке передачи кадра интервал [0, CW], из которого выбирается номер слота, удваивается. Если, например, начальный размер окна выбран равным 8 (т.е. CW = 7), то после первой коллизии размер окна должен быть равен 16 (CW = 15), после второй последовательной коллизии — 32 и т.д. Начальное значение CW, в соответствии со стандартом 802.11, должно выбираться в зависимости от типа физического уровня, используемого в беспроводной локальной сети.

Как и в методе CSMA/CD, в данном методе количество неудачных попыток передачи одного кадра ограничено, но стандарт 802.11 не устанавливает точного значения этого верхнего предела. Когда верхний предел в N попыток достигнут, кадр отбрасывается, а счётчик последовательных коллизий устанавливается в нуль. Этот счётчик также устанавливается в нуль, если кадр после некоторого

количества неудачных попыток все же передаётся успешно.

В беспроводных сетях возможна ситуация, когда два устройства (A и B) удалены и не слышат друг друга, однако оба попадают в зону охвата третьего устройства C. Это так называемая *проблема скрытого терминала*. Если оба устройства A и B начнут передачу, то они принципиально не смогут обнаружить конфликтную ситуацию и определить, почему пакеты не проходят.

В режиме доступа DCF применяются меры для устранения эффекта скрытого терминала. Для этого станция, которая хочет захватить среду и в соответствии с описанным алгоритмом начинает передачу кадра в определённом слоте, вместо кадра данных сначала посылает станции назначения короткий служебный кадр RTS (Request To Send — запрос на передачу). На этот запрос станция назначения должна ответить служебным кадром CTS (Clear To Send — свободна для передачи), после чего станция-отправитель посылает кадр данных. Кадр CTS должен оповестить о захвате среды те станции, которые находятся вне зоны сигнала станции-отправителя, но в зоне досягаемости станции-получателя, т.е. являются скрытыми терминалами для станции-отправителя.

Максимальная длина кадра данных 802.11 равна 2346 байт, длина RTS-кадра — 20 байт, CTS-кадра — 14 байт. Так как RTS- и CTS-кадры гораздо короче, чем кадр данных, потери данных в результате коллизии RTS- или CTS-кадров гораздо меньше, чем при коллизии кадров данных. Процедура обмена RTS- и CTS-кадрами не обязательна. От неё можно отказаться при небольшой нагрузке сети, поскольку в такой ситуации коллизии случаются редко.

4.8.2.1.2. Централизованный режим доступа PCF. Если в сети имеется точка доступа, то может применяться централизованный режим доступа PCF, обеспечивающий приоритетное обслуживание трафика. В этом случае точка доступа играет роль арбитра среды.

Режим доступа PCF в сетях 802.11 сосуществует с режимом DCF. Оба режима координируются с помощью трёх типов межкадровых интервалов.

После освобождения среды каждая станция отсчитывает время простоя среды, сравнивая его с тремя значениями:

— короткий межкадровый интервал (Short IFS, SIFS);
— межкадровый интервал режима PCF (PIFS);
— межкадровый интервал режима DCF (DIFS).

Захват среды с помощью распределённой процедуры DCF возможен только в том случае, когда среда свободна в течение времени, равного или большего, чем DIFS. То есть в качестве IFS в режиме DCF нужно использовать интервал DIFS — самый длительный период из трёх возможных, что даёт этому режиму самый низкий приоритет.

Межкадровый интервал SIFS имеет наименьшее значение, он служит для первоочередного захвата среды ответными CTS-кадрами или квитанциями, которые продолжают или завершают уже начавшуюся передачу кадра.

Значение межкадрового интервала PIFS больше, чем SIFS, но меньше, чем DIFS. Промежутком времени между завершением PIFS и DIFS пользуется арбитр среды. В этом промежутке он может передать специальный кадр, который говорит всем станциям, что начинается контролируемый период. Получив этот кадр, станции, которые хотели бы воспользоваться алгоритмом DCF для захвата среды, уже не могут этого сделать, они должны дожидаться окончания контролируемого периода. Его длительность объявляется в специальном кадре, но этот период может закончиться и раньше, если у станций нет чувствительного к задержкам

трафика. В этом случае арбитр передаёт служебный кадр, после которого по истечении интервала DIFS начинает работать режим DCF.

На управляемом интервале реализуется централизованный метод доступа PCF. Арбитр выполняет процедуру опроса, чтобы по очереди предоставить каждой такой станции право на использование среды, направляя ей специальный кадр. Станция, получив такой кадр, может ответить другим кадром, который подтверждает приём специального кадра и одновременно передаёт данные (либо по адресу арбитра для транзитной передачи, либо непосредственно станции).

Для того чтобы какая-то доля среды всегда доставалась асинхронному трафику, длительность контролируемого периода ограничена. После его окончания арбитр передаёт соответствующий кадр, и начинается неконтролируемый период.

Каждая станция может работать в режиме PCF, для этого она должна подписаться на данную услугу при присоединении к сети.

4.8.2.2. Типы кадров MAC

4.8.2.2.1. Контрольные кадры. Способствуют надёжной доставке информационных кадров. Существует шесть подтипов контрольных кадров:

- *Опрос после выхода из экономичного режима (PS-опрос)*. Данный кадр передаётся любой станцией станции, включающей точку доступа. В кадре запрашивается передача кадра, прибывшего, когда станция находилась в режиме энергосбережения, и в данный момент размещённого в буфере точки доступа.
- *Запрос передачи (RTS)*. Данный кадр является первым из четвёрки, используемой для обеспечения надёжной передачи данных. Станция, пославшая это сообщение, предупреждает адресата и остальные станции, способные принять данное сообщение, о своей попытке передать адресату информационный кадр.
- *Готов к передаче (CTS)*. Второй кадр четырёхкадровой схемы. Передаётся станцией-адресатом станции-источнику и предоставляет право отправки информационного кадра.
- *Подтверждение (ACK)*. Подтверждение успешного приёма предыдущих данных, кадра управления или кадра PS-опроса.
- *Без состязания (CF-конец)*. Объявляет конец периода без состязания; часть стратегии использования распределённого режима доступа.
- *CF-конец + CF-подтверждение*. Подтверждает кадр CF-конец. Данный кадр завершает период без состязания и освобождает станции от ограничений, связанных с этим периодом.

4.8.2.2.2. Информационные кадры. Существует восемь подтипов информационных кадров, собранных в две группы. Первые четыре подтипа определяют кадры, переносящие данные высших уровней от исходной станции к станции-адресату.

- *Данные*. Просто информационный кадр. Может использоваться как в период состязания, так и в период без состязания.
- *Данные + CF-подтверждение*. Может передаваться только в период без состязания. Помимо данных, в этом кадре имеется подтверждение полученной ранее информации.

— *Данные + CF-опрос*. Используется точечным координатором для доставки данных к мобильной станции и для запроса у мобильной станции информационного кадра, который находится в её буфере.
— *Данные + CF-подтверждение + CF-опрос*. Объединяет в одном кадре функции двух описанных выше кадров.

Остальные четыре подтипа информационных кадров фактически не переносят данные пользователя.

— Информационный кадр *нулевая функция* не переносит ни данных, ни запросов, ни подтверждений. Он используется только для передачи точке доступа бита управления питанием в поле управления кадром, указывая, что станция перешла в режим работы с пониженным энергопотреблением.
— Оставшиеся три кадра (*CF-подтверждение*, *CF-опрос*, *CF-подтверждение + CF-опрос*) имеют те же функции, что и описанные выше подтипы кадров (*данные + CF-подтверждение*, *данные + CF-опрос*, *данные + CF-подтверждение + CF-опрос*), но не несут пользовательских данных.

4.8.2.2.3. Кадры управления. Кадры управления используются для управления связью станций и точек доступа.

— *Запрос ассоциации*. Посылается станцией к точке доступа с целью запроса ассоциации с данной сетью с *базовым набором услуг (Basic Service Set, BSS)*. Кадр включает информацию о возможностях, например, будет ли использоваться шифрование или способна ли станция отвечать при опросе.
— *Ответ на запрос ассоциации*. Возвращается точкой доступа и указывает, что запрос ассоциации принят.
— *Запрос повторной ассоциации*. Посылается станцией при переходе между BSS, когда требуется установить ассоциацию с точкой доступа в новом BSS. Использование повторной ассоциации, а не просто ассоциации, позволяет новой точке доступа договариваться со старой о передаче информационных кадров по новому адресу.
— *Ответ на запрос повторной ассоциации*. Возвращается точкой доступа и указывает, что запрос повторной ассоциации принят.
— *Пробный запрос*. Используется станцией для получения информации от другой станции или точки доступа. Кадр используется для локализации BSS стандарта IEEE 802.11.
— *Ответ на пробный запрос*. Отклик на пробный запрос.
— *Сигнальный кадр*. Передаётся периодически, позволяет мобильным станциям локализовать и идентифицировать BSS.
— *Объявление наличия трафика*. Посылается мобильной станцией с целью уведомления других (которые могут находиться в режиме пониженного энергопотребления), что в буфере данной станции имеются кадры, адресованные другим.
— *Разрыв ассоциации*. Используется станцией для аннулирования ассоциации.
— *Аутентификация*. Для аутентификации станций используются множественные кадры.
— *Отмена аутентификации*. Передаётся для прекращения безопасного соединения.

4.8.2.3. Подстандарты

Стандарт IEEE 802.11b благодаря высокой скорости передачи данных, практически эквивалентной пропускной способности обычных проводных локальных сетей Ethernet, а также ориентации на диапазон 2,4 ГГц, завоевал наибольшую популярность у производителей оборудования для беспроводных сетей.

Поскольку оборудование, работающее на максимальной скорости 11 Мбит/с, имеет меньший радиус действия, чем на более низких скоростях, то стандартом 802.11b предусмотрено автоматическое снижение скорости при ухудшении качества сигнала.

Стандарт IEEE 802.11a имеет наибольшую ширину полосы пропускания из семейства стандартов 802.11 при скорости передачи данных до 54 Мбит/с.

В отличие от базового стандарта, ориентированного на область частот 2,4 ГГц, спецификациями 802.11a предусмотрена работа в диапазоне 5 ГГц. В качестве метода модуляции сигнала выбрано ортогональное частотное мультиплексирование (OFDM).

К недостаткам 802.11a относятся более высокая потребляемая мощность радиопередатчиков для частот 5 ГГц, а также меньший радиус действия.

Стандарт IEEE 802.11g является логическим развитием 802.11b и предполагает передачу данных в том же частотном диапазоне. Кроме того, стандарт 802.11g полностью совместим с 802.11b, т.е. любое устройство 802.11g должно поддерживать работу с устройствами 802.11b. Максимальная скорость передачи в стандарте 802.11g составляет 54 Мбит/с.

При разработке стандарта 802.11g рассматривались две отчасти конкурирующие технологии: *метод ортогонального частотного разделения OFDM* и *метод двоичного пакетного свёрточного кодирования PBCC*, опционально реализованный в стандарте 802.11b. В результате стандарт 802.11g содержит компромиссное решение: в качестве базовых применяются технологии OFDM и CCK, а опционально предусмотрено использование технологии PBCC.

Набор стандартов 802.11 определяет целый ряд технологий реализации *физического уровня (Physical Layer Protocol, PHY)*:

- уровень PHY стандарта 802.11 со скачкообразной перестройкой частоты (FHSS) в диапазоне 2,4 ГГц;
- уровень PHY стандарта 802.11 с расширением спектра методом прямой последовательности (DSSS) в диапазоне 2,4 ГГц;
- уровень PHY стандарта 802.11b с комплементарным кодированием в диапазоне 2,4 ГГц;
- уровень PHY стандарта 802.11a с ортогональным частотным мультиплексированием (OFDM) в диапазоне 5 ГГц;
- расширенный физический уровень (Extended Rate Physical Layer, ERP) стандарта 802.11g в диапазоне 2,4 ГГц.

Каждый из физических уровней стандарта 802.11 имеет два подуровня:

- *процедуру определения состояния физического уровня* (Physical Layer Convergence Procedure, PLCP);
- *подуровень физического уровня, зависящий от среды передачи* (Physical Medium Dependent, PMD).

Подуровень PLCP, по существу, является уровнем обеспечения взаимодействия, на котором осуществляется перемещение *элементов данных протокола MAC (MAC Protocol Data Units, MPDU)* между MAC-станциями с использованием подуровня PMD, на котором реализуется тот или иной метод передачи и

приёма данных через беспроводную среду. Подуровни PLCP и PMD отличаются для разных вариантов стандарта 802.11.

Одна из особенностей, лежащих в основе современных передатчиков, благодаря которой данные можно передавать с высокой скоростью, — это предположение о том, что данные, которые предлагаются для передачи, поступают, с точки зрения передатчика, случайным образом. Без этого предположения многие преимущества, получаемые за счёт применения остальных составляющих физического уровня, остались бы нереализованными.

Скрэмблирование (перестановка элементов) — метод, посредством которого принимаемые данные делаются более похожими на случайные; достигается это путём перестановки битов последовательности таким образом, чтобы превратить её из структурированной в похожую на случайную. Дескрэмблер приёмника затем выполняет обратное преобразование этой случайной последовательности с целью получения исходной структурированной последовательности. Большинство способов скрэмблирования относится к числу самосинхронизирующихся; это означает, что дескрэмблер способен самостоятельно синхронизироваться со скрэмблером.

Исходный стандарт 802.11 определяет три метода передачи на физическом уровне:

— передачу в диапазоне инфракрасных волн;
— технологию расширения спектра путём скачкообразной перестройки частоты (FHSS) в диапазоне 2,4 ГГц;
— технологию широкополосной модуляции с расширением спектра методом прямой последовательности (DSSS) в диапазоне 2,4 ГГц.

4.8.2.3.1. Передача в диапазоне инфракрасных волн. Средой передачи являются инфракрасные волны диапазона 850 нм, которые генерируются либо полупроводниковым лазерным диодом, либо светодиодом (LED). Область покрытия LAN ограничивается зоной прямой видимости. Стандарт предусматривает три варианта распространения излучения: ненаправленную антенну, отражение от потолка и фокусное направленное излучение. В первом случае узкий луч рассеивается с помощью системы линз. Фокусное направленное излучение предназначено для организации двухточечной связи.

4.8.2.3.2. Беспроводные локальные сети со скачкообразной перестройкой частоты (FHSS). Беспроводные локальные сети FHSS поддерживают скорости передачи 1 и 2 Мбит/с. Устройства FHSS делят предназначенную для их работы полосу частот от 2,402 до 2,480 ГГц на 79 неперекрывающихся каналов. Ширина каждого из 79 каналов составляет 1 МГц, поэтому беспроводные локальные сети FHSS используют относительно высокую скорость передачи символов (1 МГц) и намного меньшую скорость перестройки с канала на канал.

Последовательность перестройки частоты должна иметь следующие параметры: частота перескоков не менее 2,5 раз в секунду как минимум между шестью (6 МГц) каналами. Чтобы минимизировать число коллизий между перекрывающимися зонами покрытия, возможные последовательности перескоков должны быть разбиты на три набора последовательностей, длина которых для Северной Америки и большей части Европы составляет 26.

По сути, схема скачкообразной перестройки частоты обеспечивает медленный переход с одного возможного канала на другой таким образом, что после каждого

скачка покрывается полоса частот, равная как минимум 6 МГц, благодаря чему в многосотовых сетях минимизируется возможность возникновения коллизий.

В спецификации стандарта 802.11 оговорено использование и другого физического уровня — на основе технологии широкополосной модуляции с расширением спектра методом прямой последовательности (DSSS). Как было указано в стандарте 802.11 разработки 1997 г., технология DSSS поддерживает скорости передачи 1 и 2 Мбит/с.

4.8.2.3.3. IEEE 802.11b. Накладные расходы в этом стандарте выше, чем в проводной сети Ethernet. Поэтому крайне важно обеспечить высокую скорость передачи данных в канале. Повысить пропускную способность канала с заданной шириной полосы частот можно, разрабатывая и применяя новые методы модуляции. По этому пути пошла группа разработчиков IEEE 802.11b.

Изначально стандарт IEEE 802.11 предусматривал работу в режиме DSSS с использованием так называемой Баркеровской последовательности (Barker) длиной 11 бит: B1 = (0b10110111000). Каждый информационный бит замещается своим произведением по модулю 2 (XOR) с данной последовательностью. В результате бит заменяется последовательностью 11 чипов. Далее сигнал кодируется посредством дифференциальной двух- или четырёхпозиционной фазовой модуляции (DBPSK или DQPSK, один или два чипа на символ соответственно). При частоте модуляции несущей 11 МГц общая скорость составляет, в зависимости от типа модуляции, 1 и 2 Мбит/с.

Стандарт IEEE 802.11b дополнительно предусматривает скорости передачи 11 и 5,5 Мбит/с. Для этого используется так называемая *CCK-модуляция (Complementary Code Keying — кодирование комплементарным кодом)*.

Стандарт IEEE 802.11a появился практически одновременно с IEEE 802.11b, в сентябре 1999 г. Эта спецификация была ориентирована на работу в диапазоне 5 ГГц и основана на принципиально ином, чем описано выше, механизме кодирования данных — на частотном мультиплексировании посредством ортогональных несущих (OFDM).

Стандарт 802.11a определяет характеристики оборудования, применяемого в офисных или городских условиях, когда распространение сигнала происходит по многолучевым каналам из-за множества отражений.

В IEEE 802.11a каждый кадр передаётся посредством 52 ортогональных несущих, каждая с шириной полосы порядка 300 КГц (20 МГц/64). Ширина одного канала — 20 МГц. Несущие модулируют посредством BPSK, QPSK, а также 16- и 64-позиционной квадратурной амплитудной модуляции (QAM). В совокупности с различными скоростями кодирования (1/2 и 3/4, для 64-QAM — 2/3 и 3/4) образуется набор скоростей передачи 6, 9, 12, 18, 24, 36, 48 и 54 Мбит/с. Из 52 несущих 48 предназначены для передачи информационных символов, остальные 4 — служебные.

4.8.2.3.4. IEEE 802.11g. Стандарт IEEE 802.11g по сути представляет собой перенесение схемы модуляции OFDM, зарекомендовавшей себя в 802.11a, из диапазона 5 ГГц в область 2,4 ГГц при сохранении функциональности устройств стандарта 802.11b. Это возможно, поскольку в стандартах 802.11 ширина одного канала в диапазонах 2,4 и 5 ГГц схожа — 22 МГц.

Одним из основных требований к спецификации 802.11g была обратная совместимость с устройствами 802.11b. В качестве основного способа модуляции

принята схема *CCK (Complementary Code Keying)*, а в качестве дополнительной возможности допускается модуляция PBSS.

Разработчики 802.11g предусмотрели CCK-модуляцию для скоростей вплоть до 11 Мбит/с и OFDM для более высоких скоростей. Но сети стандарта 802.11 при работе используют принцип CSMA/CA — множественный доступ к каналу связи с контролем несущей и предотвращением коллизий. Ни одно устройство 802.11 не должно начинать передачу, пока не убедится, что эфир в его диапазоне свободен от других устройств. Если в зоне слышимости окажутся устройства 802.11b и 802.11g, причём обмен будет происходить между устройствами 802.11g посредством OFDM, то оборудование 802.11b просто не поймёт, что другие устройства сети ведут передачу, и попытается начать трансляцию. Чтобы не допустить подобной ситуации, предусмотрена возможность работы в смешанном режиме — CCK-OFDM.

Одна из основных проблем стандарта — как обеспечить бесконфликтную работу смешанных сетей 802.11b/g. Основной принцип работы в сетях 802.11 — «слушать, прежде чем вещать». Но устройства 802.11b не способны услышать устройства 802.11g в OFDM-режиме. Ситуация аналогична проблеме скрытых станций: два устройства удалены настолько, что не слышат друг друга и пытаются обратиться к третьему, которое находится в зоне слышимости обоих. Для предотвращения конфликтов в подобной ситуации в 802.11 введён защитный механизм, предусматривающий перед началом информационного обмена передачу короткого кадра *запрос на передачу* (RTS) и получение кадра подтверждения *можно передавать* (CTS). Механизм RTS/CTS применим и к смешанным сетям 802.11b/g. Естественно, эти кадры должны транслироваться в режиме CCK, который обязаны понимать все устройства. Однако защитный механизм существенно снижает пропускную способность сети.

4.8.3. Стек протоколов IEEE 802.16 (WiMAX)

WiMax (Worldwide Interoperability for Microwave Access) — стандарт беспроводной связи IEEE 802.16 (рабочая группа создана в 1999 г.).

При зазработке ставились следующие задачи:

— обеспечение доступа к услугам информационных и коммуникационных технологий для небольших поселений, удалённых регионов, изолированных объектов;
— обеспечение доступа к услугам информационных и коммуникационных технологий более половины населения планеты.

Преимущества технологии:

— стандарт объединяет технологии как уровня оператора связи, так и технологии «последней мили»;
— беспроводные технологии более гибки и, как следствие, проще в развёртывании, так как по мере необходимости могут масштабироваться;
— простота установки как фактор уменьшения затрат на развёртывание сетей в развивающихся странах, малонаселенных или удалённых районах;
— на данный момент большинство беспроводных технологий широкополосной передачи данных требуют наличия прямой видимости между объектами сети; WiMAX благодаря использованию технологии OFDM создаёт зоны покрытия в условиях отсутствия прямой видимости от клиентского оборудования до базовой станции (диаметр соты порядка нескольких километров);

— изначально содержит протокол IP, что позволяет легко и прозрачно интегрировать её в локальные сети;
— подходит для фиксированных, перемещаемых и подвижных объектов сетей на единой инфраструктуре.

Характеристики:

— пропускная способность до 135 Мбит/с при полосе несущей 28 МГц;
— доступ к среде адаптивный, динамический;
— управление сетью централизованное.

4.8.3.1. Принципы работы WiMAX

Соединение между базовой станцией и клиентским приёмником производится в СВЧ-диапазоне 2–11 ГГц. Такое соединение в идеальных условиях позволяет передавать данные со скоростью до 20 Мбит/с и не требует, чтобы станция находилась на расстоянии прямой видимости от пользователя. Этот режим работы базовой станции WiMAX близок широко распространённому стандарту IEEE 802.11, что допускает совместимость уже выпущенных клиентских устройств и WiMAX.

Между соседними базовыми станциями устанавливается постоянное соединение с использованием сверхвысокой частоты 10–66 ГГц радиосвязи *прямой видимости*. Такое соединение в идеальных условиях позволяет передавать данные со скоростью до 120 Мбит/с.

Как минимум одна из базовых станций может быть постоянно связана с сетью провайдера через широкополосное скоростное соединение. Даже при небольшом количестве точек система способна корректно распределить нагрузку за счёт сотовой топологии.

На базе сотового принципа разрабатываются также пути построения оптимальной сети, огибающей крупные объекты, когда серия последовательных станций передаёт данные по эстафетному принципу. По структуре сети стандарта IEEE 802.16 очень похожи на традиционные сети мобильной связи: здесь тоже имеются базовые станции, которые действуют в радиусе до 50 км, при этом их также необязательно устанавливать на вышках. Для них вполне подходят крыши домов, требуется лишь соблюдение условия прямой видимости между станциями. Для соединения базовой станции с пользователем необходимо наличие абонентского оборудования. Далее сигнал может поступать по стандартному Ethernet-кабелю как непосредственно на конкретный компьютер, так и на точку доступа стандарта IEEE 802.11 или в локальную проводную сеть стандарта Ethernet, что позволяет сохранить существующую инфраструктуру районных или офисных локальных сетей при переходе с кабельного доступа на WiMAX.

4.8.3.2. Режимы работы

Стандарт 802.16e-2005[1] вобрал в себя все ранее выходившие версии и на данный момент предоставляет следующие режимы:

— стационарный доступ (Fixed WiMAX) (рабочая группа IEEE 802.16d, стандарт IEEE 802.16-2004);

[1] Документ IEEE 802.16e-2005 сам по себе является не стандартом, а дополнением к стандарту IEEE 802.16-2004.

— сеансовый доступ (Nomadic[1] WiMAX);
— доступ в режиме перемещения (Portable WiMAX);
— мобильный доступ (Mobile WiMAX) (дополнение IEEE 802.16e-2005).

4.8.3.2.1. Fixed WiMAX. Фиксированный доступ представляет собой альтернативу широкополосным проводным технологиям (xDSL, T1). Стандарт использует диапазон частот 10–66 ГГц. Этот частотный диапазон из-за сильного затухания коротких волн требует прямой видимости между передатчиком и приёмником сигнала, но позволяет избежать одной из главных проблем радиосвязи — многолучевого распространения сигнала. При этом ширина каналов связи в этом частотном диапазоне довольно велика (типичное значение — 25 или 28 МГц), что позволяет достигать скоростей передачи до 120 Мбит/с. Фиксированный режим включался в версию стандарта IEEE 802.16d-2004.

4.8.3.2.2. Nomadic WiMAX. Сеансовый доступ добавил понятие сессий к уже существующему Fixed WiMAX. Наличие сессий позволяет свободно перемещать клиентское оборудование между сессиями и восстанавливать соединение уже с помощью других вышек WiMAX, нежели те, что использовались во время предыдущей сессии. Такой режим разработан в основном для портативных устройств. Введение сессий позволяет также уменьшить расход энергии клиентского устройства.

4.8.3.2.3. Portable WiMAX. Для режима Portable WiMAX добавлена возможность автоматического переключения клиента от одной базовой станции WiMAX к другой без потери соединения. Однако для данного режима все ещё ограничена скорость передвижения клиентского оборудования — 40 км/ч.

4.8.3.2.4. Mobile WiMAX. Этот режим был разработан в стандарте 802.16e-2005 и позволил увеличить скорость перемещения клиентского оборудования до 120 км/ч. Основные достижения этого режима:
— устойчивость к многолучевому распространению сигнала и собственным помехам;
— масштабируемая пропускная способность канала;
— технология *Time Division Duplex (TDD)*, которая позволяет эффективно обрабатывать асимметричный трафик и упрощает управление сложными системами антенн за счёт эстафетной передачи сессии между каналами;
— технология *Hybrid-Automatic Repeat Request (H-ARQ)*, которая позволяет сохранять устойчивое соединение при резкой смене направления движения клиентского оборудования;
— распределение выделяемых частот и использование субканалов при высокой загрузке позволяет оптимизировать передачу данных с учётом силы сигнала клиентского оборудования;
— управление энергосбережением позволяет оптимизировать затраты энергии на поддержание связи портативных устройств в режиме ожидания или простоя;
— технология *Network-Optimized Hard Handoff (HHO)*, которая позволяет до 50 мс и менее сократить время на переключение клиента между каналами;

[1] Буквально: кочующий.

- технология *Multicast and Broadcast Service (MBS)*, которая объединяет функции DVB-H, MediaFLO и 3GPP E-UTRA для:
  - достижения высокой скорости передачи данных с использованием одночастотной сети;
  - гибкого распределения радиочастот;
  - низкого потребления энергии портативными устройствами;
  - быстрого переключения между каналами;
- технология *Smart Antenna*, поддерживающая субканалы и эстафетную передачу сессии между каналами, что позволяет использовать сложные системы антенн, включая формирование диаграммы направленности, пространственно-временное маркирование, пространственное мультиплексирование;
- технология *Fractional Frequency Reuse*, которая позволяет контролировать наложение/пересечение каналов для повторного использования частот с минимальными потерями;
- размер фрейма в 5 мс обеспечивает компромисс между надёжностью передачи данных за счёт использования малых пакетов и накладными расходами посредством увеличения числа пакетов (и, как следствие, заголовков).

4.8.4. Технология Bluetooth

Bluetooth представляет собой беспроводную технологию, обеспечивающую беспроводную передачу данных на небольших расстояниях между различными устройствами (например, мобильными персональными компьютерами, мобильными телефонами и другими устройствами) в режиме реального времени. При этом возможна передача как цифровых данных, так и звуковых сигналов.

Работа над концепцией системы Bluetooth началась в 1994 г. шведской компанией Ericsson. В феврале 1998 г. по инициативе пяти ведущих зарубежных компаний — Ericsson, IBM, Intel, Nokia и Toshiba — была организована специальная группа (Special Interest Group, SIG) [6], в задачи которой входило продвижение этой технологии. В мае того же года последовало объявление об учреждении концерна Bluetooth. В конце 1999 г. появились первые спецификации на соответствующее оборудование, ставшие впоследствии стандартом де-факто. На спецификациях Bluetooth v. 1.x базируется стандарт IEEE 802.15.1 [7], утверждённый в 2002 г.

Устройства версий 1.0 и 1.0B имели плохую совместимость между продуктами различных производителей. Основным недостатком была невозможность реализовать анонимность на протокольном уровне.

В Bluetooth 1.1 было исправлено множество ошибок, найденных в 1.0B, добавлена поддержка для нешифрованных каналов, *индикация уровня мощности принимаемого сигнала (Received Signal Strength Indicator, RSSI)*.

В версии 1.2 (2003 г.) была добавлена *технология адаптивной перестройки рабочей частоты (Adaptive Frequency-Hopping Spread Spectrum, AFH)*, что улучшило сопротивляемость к электромагнитной интерференции (помехам) путём использования разнесённых частот в последовательности перестройки. Также увеличилась скорость передачи и добавилась технология *eSCO (Extended Synchronous Connections)*, которая улучшила качество передачи голоса путём повторения повреждённых пакетов. В *HCI (Host Controller Interface)* добавилась поддержка трёхпроводного интерфейса *UART (Universal Asynchronous Receiver/-Transmitter)*.

Основные отличия Bluetooth 1.2: ускоренное установление соединения, адаптивная схема переключения каналов (от 20 до 79), усовершенствованные алгоритмы передачи данных.

Версия Bluetooth 2.0+EDR (2004 г.) состоит из двух частей, которые могут поддерживаться аппаратурой независимо: обновлённая версия спецификации Bluetooth (без принципиальных отличий от версии 1.2) и *расширенный набор скоростей передачи данных (Enhanced Data Rate, EDR)*. В режиме EDR применяется дифференциальная фазовая модуляция, увеличивающая базовую скорость передачи с 1 до 3 Мбит/с.

Стандарт Bluetooth 2.0+EDR полностью совместим с Bluetooth 1.0 и 1.2; скорость передачи в пикосети не ограничивается скоростью самого медленного.

В 2007 г. появилась обновлённая версия Bluetooth 2.1 (полное название Bluetooth Core Specification Version 2.1 + EDR). Эта версия полностью совместима с версией 2.0. В ней удалось снизить энергопотребление, а также усовершенствовать алгоритм связи.

4.8.4.1. Основные параметры радиоинтерфейса Bluetooth

- Диапазон частот: 2,4–2,4835 ГГц — промышленный, научный и медицинский диапазон частот (Industrial, Scientific and Medical band, ISM band)[1].
- Число несущих частот: 23–79 с разносом 1 МГц (16/32 в одной пикосети).
- Метод доступа: скачкообразная перестройка частоты и *дуплексная передача с временны́м разделением каналов (Frequency-hopping spread spectrum / Time-Division Duplex, FHSS/TDD)* (1600 скачков в секунду).
- Метод модуляции: *частотная модуляция с гауссовским сглаживанием (Gaussian Frequency Shift Keying, GFSK)* — двухуровневая схема кодирования сигнала, в которой логическому 0 и 1 соответствуют две разные частоты, коэффициент сглаживания формы входных импульсов $h = 0,35$.
- Скорость передачи по радиоканалу: 1 Мбит/с.
- Полоса пропускания: 220 кГц (по уровню 3 дБ), 1 МГц (по уровню 20 дБ).
- Мощность передатчика: 100 мВт (для связи до 100 м; 20 дБм), 2 мВт (до 10 м; 4 дБм) и 1 мВт (10 см; 0 дБм).

4.8.4.2. Принцип работы

Абонентские устройства Bluetooth объединяются в группы (пикосети), совместно использующие один радиоканал. В состав каждой пикосети входят один ведущий приёмопередатчик (с опорным генератором, который синхронизирует внутренний трафик сети) и до семи ведомых (синхронизируемых). Ведомое устройство вычисляет разность между частотами собственного и ведущего генераторов, и в процессе вхождения в синхронизм эта погрешность учитывается, что обеспечивает точное соответствие излучаемой частоты данного и ведущего устройств.

Вид псевдослучайной последовательности однозначно идентифицирует ведущий приёмопередатчик, а её фаза (псевдослучайный сдвиг) является адресным признаком ведомого устройства. Период повторения последовательности, определяющей закон перестройки частоты, достаточно большой (свыше 23 ч.). В каждой пикосети используется своя псевдослучайная последовательность, что позво-

[1] ISM band определяет полосы частот (918, 2450 и 5800 МГц, 22,5 ГГц) для работы промышленной, научной и медицинской радиослужб.

ляет множеству пикосетей одновременно работать по одному и тому же каналу связи, не создавая взаимных помех.

4.8.4.3. Синхронное и асинхронное соединения

Bluetooth имеет возможность организовывать как синхронное, так и асинхронное соединение.

Синхронное соединение (Synchronous Connection Oriented, SCO) возможно только в режиме точка–точка и применяется для передачи информации, чувствительной к задержкам (например, голоса). Основное (ведущее) устройство поддерживает до трёх синхронных соединений, подчинённое — до трёх синхронных соединений с одним основным устройством или до двух — с разными основными устройствами. При синхронном соединении основное устройство резервирует временны́е сегменты, следующие через так называемые SCO-интервалы. Даже если пакет принят с ошибкой, повторно при синхронном соединении он не передаётся.

Асинхронное соединение (Asynchronous Connection Less, ACL) возможно между основным и всеми активными подчинёнными устройствами в пикосети (режим точка–многоточка). Основное и подчинённое устройства могут поддерживать только одно асинхронное соединение. Подчинённое устройство отправляет пакет основному, только если в предыдущем временно́м интервале на его адрес пришёл пакет от основного устройства. Асинхронное соединение позволяет повторно передавать пакеты, принятые с ошибками.

Таким образом, Bluetooth может поддерживать один асинхронный канал данных (со скоростью до 723,2 Кбит/с в прямом и 57,6 Кбит/с в обратном направлениях), до трёх синхронных (с постоянной скоростью 64 Кбит/с в каждом направлении) голосовых каналов или канал с одновременной асинхронной передачей данных и синхронной передачей голоса (со скоростью до 433,9 Кбит/с в каждом направлении).

4.8.4.4. Структура кадра

Стандартный кадр Bluetooth содержит *код доступа (Access Code)* (длина 72 бита), *заголовок* (длина 54 бита) и *информационное поле* (длина не более 2745 бит) (рис. 4.21).

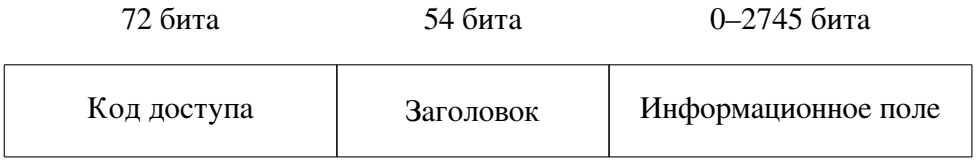


Рис. 4.21. Стандартный кадр Bluetooth

Код доступа идентифицирует пакеты, принадлежащие одной пикосети, а также используется для синхронизации и процедуры запросов; состоит из трёх полей:

— преамбулы (Preamble) (4 бита),
— кода синхронизации (Sync Word) (64 бита),
— концевика (Trailer) (4 бита контрольной суммы).

Заголовок (рис. 4.22) содержит информацию для управления связью и состоит из шести полей:

— поле *Адрес (AM_ADDR)* (3 бита) — содержит MAC-адрес узла назначения;
— поле *Тип пакета (TYPE)* (4 бита) — указывает код одного из 12 типов данных (ACL, SCO, опрос или пустой кадр), метод коррекции ошибок и число временных интервалов, из которых состоит кадр;
— поле *Поток (FLOW)* (1 бит) — осуществляет управление потоком данных, показывает готовность устройства к приёму;
— поле *Признак повторной передачи (ARQ)* (1 бит) — определяет корректность приёма;
— поле *SEQN* (1 бит) — служит для определения последовательности пакетов;
— поле *Контроль ошибок в заголовке (Header Error Check, HEC)* (8 бит) — контрольная сумма.

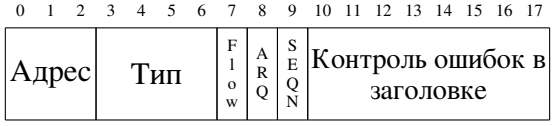


Рис. 4.22. Заголовок Bluetooth

Информационное поле имеет три сегмента:

— поле *Заголовок полезной информации* (8 бит) определяет логический канал, управление потоком в логических каналах, а также имеет указатель длины полезной информации;
— поле *Тело полезной информации* (0–2721 бит) включает пользовательскую информацию; длина этого сегмента указана в поле длины заголовка полезной информации;
— поле *Циклический избыточный код (Cyclic Redundancy Check, CRC)* (16 бит) служит для контроля целостности передаваемых данных.

Существует шесть типов пакетов Bluetooth.

4.8.4.5. Протоколы Bluetooth

Иерархия протоколов Bluetooth не соответствует моделям ISO/OSI, TCP/IP и IEEE 802. В спецификации определено 5 уровней: физический, базовый, управления каналом, сетевой и уровень приложений.

На *физическом уровне* определены параметры радиоинтерфейса Bluetooth.

Базовый уровень (baseband) и *уровень управления связью (Link Control Layer)* обеспечивают физическую радиочастотную связь между устройствами Bluetooth, образующими пикосеть.

На базовом уровне определено 13 типов пакетов. Пакеты ID, NULL, POLL, FHS , DM1 определены как для синхронных, так и для асинхронных соединений. Пакеты DH1, AUX1, DM3, DH3, DM5 и DH5 определены только для асинхронного соединения. Форматы пакетов HV1, HV2, HV3 и DV определены только для синхронного соединения. Кроме того, на данном уровне определены пять логических каналов: *LC (Control Channel)* и *LM (Link Manager)* используются на канальном уровне, а *UA (User Asynchronous)*, *UI (User Isosynchronous)* и *US (User Synchronous)* служат для асинхронной, изосинхронной и синхронной транспортировки пользовательских данных.

Протокол управления связью (Link Manager Protocol, LMP) отвечает за установление подключений между устройствами Bluetooth. Также сюда относятся и вопросы безопасности, такие как идентификация и шифрование, связанные с генерированием ключей шифрования и подключения, а также с обменом ключами и их проверкой. Кроме того, протокол контролирует режимы питания и исполнительные циклы устройств Bluetooth, а также состояние подключения того или иного устройства к пикосети.

Протокол управления логическим подключением и адаптацией (Logical Link Control and Adaptation Protocol, L2CAP) адаптирует протоколы верхнего уровня над Baseband. L2CAP является базовым протоколом передачи данных для Bluetooth. L2CAP работает только с ACL-соединениями. Многие протоколы и службы более высокого уровня используют L2CAP как транспортный протокол.

Протокол обнаружения услуг (Service Discovery Protocol, SDP) использует L2CAP в качестве транспортного протокола, что позволяет запросить информацию о самом устройстве, его услугах и характеристиках этих услуг, а после этого может быть установлено соединение между двумя или несколькими устройствами Bluetooth.

Протокол замены кабеля (RFCOMM) также использует L2CAP в качестве транспортного протокола. Протокол RFCOMM эмулирует соединение PPP (point-to-point) по последовательному порту, обеспечивает транспортировку при выполнении услуг верхнего уровня.

Двоичный протокол управления телефонией (Telephony Control Protocol Specification Binary, TCS Binary) является биториентированным протоколом и определяет контроль сигнализации вызова для установления речевого вызова или вызова данных между устройствами Bluetooth. Кроме того, он определяет процедуры управления мобильностью при манипулировании с группами TCS-устройств Bluetooth.

4.8.4.6. Профили Bluetooth

Профили Bluetooth представляют собой общие механизмы (протоколы и функции), через которые доступные устройства Bluetooth взаимодействуют с другими устройствами. Профили определяют области возможного применения устройства Bluetooth. Если устройства от различных производителей соответствуют одному профилю, определённому в спецификации Bluetooth, они смогут взаимодействовать друг с другом.

— *Профиль общего доступа (Generic Access Profile, GAP)*
Профиль GAP отвечает за поддержание связи между устройствами, выявление других доступных профилей, а также за безопасность соединений. Этот профиль должен быть включён во все устройства Bluetooth. В него входят функции, необходимые для работы всех основных протоколов Bluetooth.

— *Профиль последовательного порта (Serial Port Profile, SPP)*
Профиль SPP позволяет устройствам Bluetooth эмулировать последовательный порт при помощи протокола RFCOMM. Профиль SPP определяет, каким образом два доступных устройства Bluetooth будут осуществлять обмен данными посредством эмуляции интерфейса RS-232 или интерфейса USB.

— *Профиль приложения обнаружения услуг (Service Discovery Application Profile, SDAP)*

Профиль SDAP описывает, каким образом приложение должно использовать *протокол обнаружения услуг (Service Discovery Protocol, SDP)*. Профиль SDAP необходим для того, чтобы любое приложение имело возможность узнать, какие услуги (сервисы) Bluetooth является доступными на любом устройстве Bluetooth, с которым оно соединено.

— *Общий профиль обмена объектами (Generic Object Exchange Profile, GOEP)*
Профиль GOEP используется для непосредственного (без использования IP) обмена объектами между двумя устройствами. Объект может иметь любой тип, например, изображение, документ, визитная карточка и т.д. Профиль определяет устройству одну из двух ролей: сервер, который определяет место, куда объект был помещён, и клиент, который инициализирует механизм передачи.

— *Профиль дозвона по сети (Dial-Up Networking Profile, DUN)*
DUN обеспечивает стандартный доступ к сети Интернет и другому сервису модемной связи по беспроводной технологии Bluetooth.

— *Профиль факсимильной связи (Fax Profile, FAX)*
Профиль FAX определяет, каким образом устройство, имеющее шлюз факсимильного аппарата, может использоваться в качестве оконечного устройства. Профиль FAX предназначен для обеспечения интерфейса между мобильным телефоном (или телефоном стационарной сети) и персональным компьютером с установленным программным обеспечением, поддерживающим факс.

— *Профиль гарнитуры (Headset Profile, HSP)*
Профиль HSP определяет способ, посредством которого Bluetooth обеспечивает беспроводное соединение устройства с гарнитурой, оснащённой динамиками и, возможно, микрофоном.

— *Профиль доступа к локальной сети (LAN Access Profile, LAP)*
Профиль LAP предназначен для создания IP-сетей и позволяет создавать небольшие беспроводные сети Intranet, объединяющие ПК или смарт-телефоны. Он также используется точками доступа для связи с кабельными сетями, будь то локальные сети или Internet.

— *Профиль передачи файлов (File Transfer Profile, FTP)*
Профиль FTP определяет, каким образом файлы на устройстве сервера могут быть просмотрены устройством клиента. Если местонахождение файла определено клиентом, то файл может быть перемещён от сервера к клиенту или помещён клиентом на сервер, используя профиль GOEP.

— *Профиль помещения объектов в стек (Open Push Profile, OPP)*
Профиль OPP управляет обменом электронными визитками в формате vCard (расширение файлов *.vcf). Эти визитки содержат ту же информацию, что и традиционные, но при этом они могут быть автоматически занесены в личную информационную систему (PIM) или в базу данных.

— *Профиль синхронизации (Synchronization Profile, SYNC)*
Профиль SYNC используется вместе с GOEP, чтобы обеспечить синхронизацию календаря и адресной информации (элементы управления персональной информации — PIM) между доступными Bluetooth-устройствами. Основное применение этого профиля — обмен данными между персональным цифровым секретарём (PDA) и компьютером.

— *Профиль беспроводной телефонной связи (Cordless Telephony Profile, CTP)*
Профиль CTP определяет, каким образом беспроводной телефон может быть использован в технологии Bluetooth. Этот профиль может использоваться

или для беспроводного телефона, или для мобильного телефона, который функционирует как беспроводной телефон вблизи от базовой станции.

— *Профиль внутренней связи (Intercom Profile, ICP)*
Этот профиль обеспечивает двустороннюю голосовую связь между устройствами Bluetooth. Он рассчитан на прямое взаимодействие двух устройств, расположенных в зоне взаимной досягаемости. Технология была разработана таким образом, чтобы, с одной стороны, не создавать ненужных помех для других пользователей, а с другой — быть невосприимчивым к радиосигналам других технологий, работающих на этих же частотах.

Дополнительные профили Bluetooth для устройств печати:

— *Профиль замены кабеля твёрдой копии (Hard Copy Cable Replacement Profile, HCRP)*
Профиль HCRP обеспечивает беспроводной вариант связи в качестве замены кабельного соединения между устройством и принтером.

— *Основной профиль принтера (Basic Printing Profile, BPP)*
Профиль BPP обеспечивает механизм формирования заданий вывода на печать текстов, сообщений электронной почты, изображений, визиток типа vCards и других объектов. Отличие этого профиля от HCRP заключается в том, что BPP не требует наличия специфических драйверов для каждого конкретного принтера.

Дополнительные профили Bluetooth для аудио- и видеоаппаратуры:

— *Общий профиль распространения аудио и видео (General Audio/Video Distribution Profile, GAVDP)*
Профиль GAVDP является основой для профилей A2DP и VDP, применяемых в системах распределения видео- и аудиопотоков, использующих беспроводную технологию Bluetooth.

— *Расширенный профиль распространения аудио (Advanced Audio Distribution Profile, A2DP)*
Профиль A2DP описывает, каким образом качественный стереозвук проходит от источника до приёмника.

— *Профиль распространения видео (Video Distribution Profile, VDP)*
Профиль VDP определяет, каким образом доступное Bluetooth-устройство обеспечивает передачу потоков видеоинформации, используя Bluetooth-технологии.

— *Профиль дистанционного управления аудио- и видеоаппаратурой (Audio/Video Remote Control Profile, AVRCP)*
Профиль AVRCP обеспечивает стандартный интерфейс для управления высококачественной аудио- и видеоаппаратурой. Использование этого профиля позволяет единственному пульту дистанционного управления осуществлять управление всей аудио- и видеоаппаратурой, которая находится в окрестности. Профиль AVRCP даёт возможность управлять характеристиками мультимедиа потоков, например, регулировкой громкости, пуском, приостановкой и остановкой плейера, а также выполнять другие подобные операции дистанционного управления.

Основную конфигурацию дополняют другие профили Bluetooth:

— *Основной профиль изображения (Basic Imaging Profile, BIP)*
Профиль BIP обеспечивает механизм дистанционного управления устройствами записи, передачи и отображения изображений (например, управление затвором цифровой фотокамеры с помощью мобильного телефона). До-

бавляется в основную конфигурацию профилей под управление профиля GOEP.

— *Профиль Hands-Free (Hands-Free Profile, HFP)*
Профиль HFP описывает, каким образом устройство-шлюз может использоваться для размещения и получения вызовов устройства hands-free. Типичный пример — применение мобильного телефона в качестве устройства-шлюза. Профиль HFP позволяет также использовать ресурсы мультимедиа персонального компьютера в качестве аппаратуры громкой связи мобильного телефона. Добавляется в основную конфигурацию профилей под управление профиля SPP.

Глава 5. Сетевой уровень

5.1. Протокол IPv4

Основой семейства протоколов TCP/IP является *межсетевой протокол IP (Internet Protocol)*, определённый в RFC 791 [8]. Протокол IP называют также *IPv4 (IP Version 4)*, чтобы отличать его от протокола IP версии 6 (IPv6 или IP Next Generation, IPng).

Протокол IP изначально разрабатывался как протокол передачи пакетов в составных сетях. Он хорошо работает в сетях со сложной топологией. Так как протокол IP является датаграммным протоколом, то он не гарантирует доставку пакетов до узла назначения.

5.1.1. Формат пакета IP

IP-пакет состоит из заголовка и поля данных. Заголовок обычно имеет длину 20 байт. Рассмотрим его структуру (рис. 5.1).

Рис. 5.1. Формат заголовка пакета IPv4

Поле *Версия (Version)* (длина 4 бита) указывает версию протокола IP. Напомним, что сейчас мы рассматриваем протокол IP версии 4 (IPv4).

Поле *Длина заголовка (IHL)* (длина 4 бита) указывает длину заголовка в 32-битных словах[1]. Обычно заголовок имеет длину 20 байт (=5 32-битовых слов), однако возможно включение в заголовок дополнительной информации, размещённой в поле *Опции*. Максимальная длина заголовка — 60 байт (=15 32-битовых слов)[2].

[1] Очевидно, что в четырёх битах можно максимально закодировать число 15.

[2] Может показаться, что для заголовка IP достаточно 60 байт в любых случаях. Однако это не так. Например, при использовании поля опции *Запись маршрута (IP Record Route)* в заголовке нельзя записать более девяти пройденных на маршруте точек.

Поле *Тип обслуживания (Type of Service, TOS)* (длина 8 бит) состоит из нескольких подполей (рис. 5.2). Вначале идёт подполе *приоритета (Precedence)* пакета (длина 3 бита). Приоритет может иметь значение от самого низкого — 0 (обычный пакет) до самого высокого — 7 (пакет управляющей информации). Более важные пакеты обрабатываются в первую очередь.

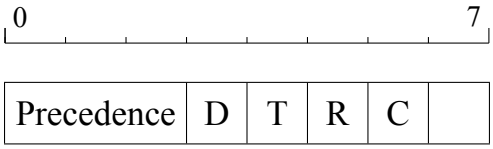


Рис. 5.2. Поле *Тип обслуживания* заголовка IP

Далее следуют четыре бита, задающие тип обслуживания. Из этих четырёх бит только один может быть выставлен в 1. Они имеют следующий смысл: *малая задержка (Low Delay, D)*, *высокая пропускная способность (High Throughput, T)*, *высокая надёжность (High Reliability, R)*, *низкая стоимость (Low Cost, C)*. Последний бит подполя был добавлен уже после появления RFC 791 [8]. Смысл значений типа обслуживания абстрактен[1].

Последний бит поля не используется.

Поле *Общая длина (Total Length)* (длина 16 бит) описывает общий размер пакета в байтах с учётом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью этого поля и составляет 65535 байт, однако обычно столь большие пакеты не используются.

Поле *Идентификатор пакета (Identification)* (длина 16 бит) определяет каждый посланный узлом пакет IP и увеличивается на 1 при отправке каждого пакета. Исключением из правил являются фрагментированные пакеты IP, в которых значение поля идентификатора пакета одинаково для всех отправленных фрагментов. При фрагментации используются также поля *Флаги* и *Смещение фрагмента*.

Поле *Флаги (Flags)* (длина 3 бита) (рис. 5.3) содержит признаки, связанные с фрагментацией. Установленный бит *Не фрагментировать (Do not Fragment, DF)* запрещает маршрутизатору фрагментировать данный пакет, а установленный бит *Есть ещё фрагменты (More Fragments, MF)* говорит о том, что данный пакет является промежуточным фрагментом. Первый бит зарезервирован.

Поле *Смещение фрагмента (Fragment Offset)* (длина 13 бит) задаёт смещение в байтах поля данных этого пакета от начала поля данных исходного фрагментированного пакета. Смещение должно быть кратным 8 байтам.

Поле *Время жизни (Time to Live, TTL)* (длина 8 бит) указывает предельный срок, в течение которого пакет может перемещаться по сети. Значение этого поля уменьшается на 1 каждую секунду или всякий раз, как пакет IP проходит через маршрутизатор. Когда значение этого поля достигает 0, маршрутизатор отбрасы-

[1] В RFC 791 отмечено: «Тип обслуживания — это абстрактный и обобщённый набор параметров, характеризующий услуги, которые предоставляются сетями, составляющими объединённую сеть. Значения из поля *Тип обслуживания* должны использовать шлюзы при выборе параметров реальной пересылки информации в данной сети, сети следующего участка на пути пакета или следующего шлюза при маршрутизации сетевой датаграммы».

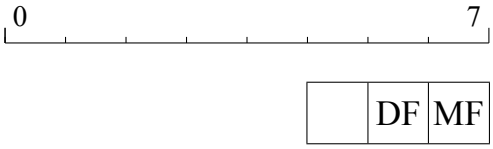


Рис. 5.3. Поле *Флаги* заголовка IP

вает пакет и отсылает отправителю пакет ICMP, уведомляя его об истечении времени жизни. Основным предназначением этого пакета является предотвращение зацикливания пакета между маршрутизаторами.

Если получатель принял не все фрагменты пакета IP, а время жизни дошло до 0, то получатель направляет отправителю пакет ICMP, уведомляющий об истечении времени жизни уже в процессе ожидания начала сборки пакета.

Поле *Протокол верхнего уровня (Protocol)* (8 бит) указывает, какому протоколу верхнего уровня принадлежит информация, размещённая в поле данных пакета. Значение идентификаторов для различных протоколов приводятся в RFC «Assigned Numbers» (табл. 5.1).

Таблица 5.1

Идентификаторы наиболее распространённых протоколов

Значение	Название	Описание
1	ICMP	Internet Control Message Protocol — протокол межсетевых управляющих сообщений
2	IGMP	Internet Group Menagement Protocol — протокол управления межсетевыми группами
3	GGP	Gateway-to-Gateway Protocol — межшлюзовый протокол
4	IP	Инкапсуляция IP в IP
6	TCP	Transmission Control Protocol — протокол управления передачей
8	EGP	Exterior Gateway Protocol — протокол внешнего шлюза
17	UDP	User Datagram Protocol — протокол пользовательских датаграмм
88	IGRP	Interior Gateway Routing Protocol — внутренний протокол маршрутизации
89	OSPF	Open Shortest Path First

Поле *Контрольная сумма (Header Checksum)* (длина 2 байта) рассчитывается только по заголовку (данные не учитываются). Поскольку некоторые поля заголовка меняют своё значение в процессе передачи пакета по сети, контрольная

сумма проверяется и повторно рассчитывается при каждой обработке заголовка пакета IP. При вычислении контрольной суммы значение самого поля *Контрольная сумма* выставляется в ноль.

Поля *IP-адрес источника (Source IP Address)* и *IP-адрес назначения (Destination IP Address)* (длина по 32 бита) содержат адреса отправителя и получателя соответственно.

Поле *Опции (Option)* является необязательным и используется обычно при отладке. Это поле состоит из нескольких подполей, каждое из которых может иметь один из восьми предопределённых типов. Так как число подполей может быть произвольным, то после этого поля может идти поле *Выравнивание (Padding)*, служащее для выравнивания заголовка по 32-битной границе. Выравнивание осуществляется нулями.

5.1.2. Схема адресации протокола IPv4

В сети IP все устройства имеют уникальный адрес (IP-адрес). IP-адрес характеризует не само устройство, а соединение устройства с сетью (например, устройство с двумя сетевыми интерфейсами будет иметь как минимум два IP-адреса). Схема адресации протокола IPv4 описана в документах RFC 990 [9], RFC 997 [10].

IP-адрес имеет длину 32 бита. Для удобства принято записывать IP-адрес в виде двоично-десятичного числа: каждый байт (октет) записывается в виде десятичного числа в диапазоне от 0 до 255; октеты разделены точками (например, 192.168.0.1). Такая форма записи носит название *десятично-точечной нотации*.

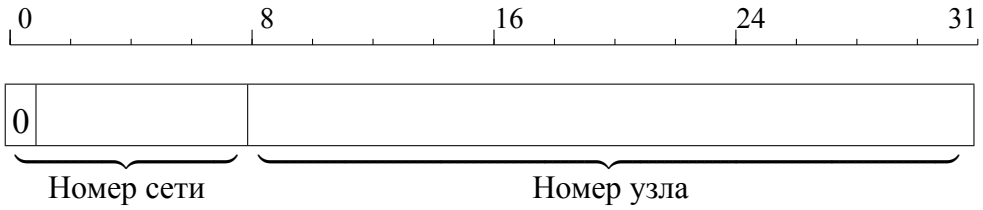
5.1.2.1. Классы адресов

IP-адреса разделяются на 5 классов: A, B, C, D, E. Адреса классов A, B и C делятся на две логические части: *номер сети* и *номер узла* (рис. 5.4, табл. 5.2).

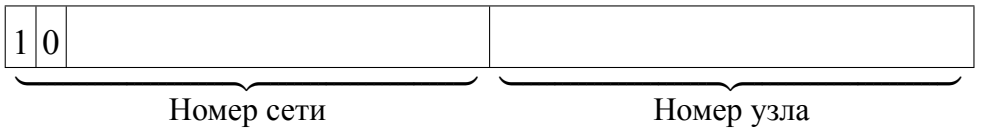
У адресов класса A старший бит установлен в 0 (рис. 5.4а). Длина сетевого префикса — 8 бит. Для номера узла выделяется 24 бита. Таким образом, в классе A может быть 126 сетей ($2^7 - 2$, поскольку два номера сети имеют специальное значение, см. п. 5.1.2.2). Каждая сеть этого класса может поддерживать максимум 16777214 узлов ($2^{24} - 2$, также см. п. 5.1.2.2). Адресный блок класса A может содержать максимум 2^{31} уникальных адресов, в то время как в протоколе IPv4 возможно существование 2^{32} адресов. Таким образом, адресное пространство класса A занимает 50% всего адресного пространства протокола IPv4. Адреса класса A предназначены для использования в больших сетях, с большим количеством узлов. На данный момент все адреса класса A распределены.

У адресов класса B два старших бита установлены в 1 и 0 соответственно (рис. 5.4б). Длина сетевого префикса — 16 бит. Поле номера узла тоже имеет длину 16 бит. Таким образом, число сетей класса B равно 16384 (2^{14}); каждая сеть класса B может поддерживать до 65534 узлов ($2^{16} - 2$). Адресный блок сетей класса B содержит 2^{30} уникальных адресов, т.е. 25% всего адресного пространства. Класс B предназначен для применения в сетях среднего размера.

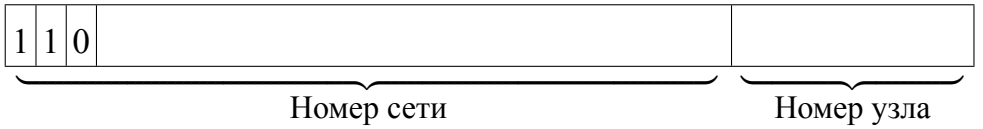
У адресов класса C три старших бита установлены в 1, 1 и 0 соответственно (рис. 5.4в). Префикс сети имеет длину 24 бита, номер узла — 8 бит. Максимально возможное количество сетей класса C составляет 2097152 (2^{21}). Каждая сеть может поддерживать максимум 254 узла ($2^8 - 2$). Весь адресный блок сетей класса C



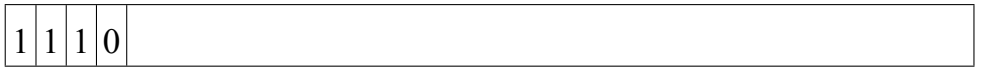
(а) Класс A



(б) Класс B



(в) Класс C



(г) Класс D

(д) Класс E

Рис. 5.4. Классы сетей IPv4

Таблица 5.2

Классы IP-адресов

Класс адреса	Формат записи	Старшие биты	Границы адресов сети	Количество битов в адресе сети/хоста	Максимальное количество сетей	Максимальное количество хостов	Назначение
A	N.H.H.H	0	от 1.0.0.0 до 126.0.0.0	8/24	$2^7 - 2$	$2^{24} - 2$	Большие организации
B	N.N.H.H	1 0	от 128.0.0.0 до 191.255.0.0	16/16	2^{14}	$2^{16} - 2$	Организации среднего размера
C	N.N.N.H	1 1 0	от 192.0.0.0 до 223.255.255.0	24/8	2^{21}	$2^8 - 2$	Малые организации
D	—	1 1 1 0	от 224.0.0.0 до 239.255.255.255	—	—	—	Групповое вещание
E	—	1 1 1 1	от 240.0.0.0 до 247.255.255.255	—	—	—	Экспериментальные

Обозначения: N — часть адреса, относящаяся к номеру сети, H — часть адреса, относящаяся к номеру узла.

содержит 2^{29} уникальных адреса, что равно 12,5% от всего адресного пространства. Класс C предназначен для сетей с небольшим количеством узлов.

Два оставшихся класса имеют другую структуру адреса.

У адресов класса D четыре старших бита установлены в 1, 1, 1 и 0 соответственно (рис. 5.4г). Адреса этого класса используются для поддержки *группового вещания (Multicasting)*. При групповом вещании пакет передаётся нескольким узлам по схеме «один-ко-многим». Адрес класса D является идентификатором такой группы. Узлы сами идентифицируют себя, определяя, к какой группе они относятся. Узлы, принадлежащие одной группе, могут быть распределены по разным сетям произвольным образом.

У адресов класса E четыре старших бита установлены в 1, 1, 1 и 1 соответственно (рис. 5.4д). Класс E зарезервирован для экспериментального использования.

5.1.2.2. Служебные IP-адреса

Некоторые IP-адреса являются зарезервированными (табл. 5.3).

Таблица 5.3

Служебные IP-адреса

Поле сети	Поле узла	Интерпретация
Все биты равны 0	Все биты равны 0	Данное устройство
Все биты равны 0	Номер узла	Устройство в данной IP-сети
Все биты равны 1	Все биты равны 1	Все устройства в данной IP-сети (ограниченное широковещательное сообщение (Limited Broadcast))
Номер сети	Все биты равны 0	Данная IP-сеть
Номер сети	Все биты равны 1	Все устройства в указанной IP-сети (широковещательное сообщение (Broadcast))
127		Возвратный адрес (Loopback)

Для таких адресов существуют соглашения об их особой интерпретации.

1) Если все биты IP-адреса установлены в нуль, то он обозначает адрес данного устройства.
2) Если в поле номера сети стоят нули, то считается, что получатель принадлежит той же самой сети, что и отправитель.
3) Если все биты IP-адреса установлены в единицу, то пакет с таким адресом должен рассылаться всем узлам, находящимся в той же сети, что и отправитель. Такая рассылка называется *ограниченным широковещательным сообщением (Limited Broadcast)*.
4) Если все биты номера узла установлены в нуль, то пакет предназначен для данной сети.

5) Если все биты в поле номера узла установлены в единицу, то пакет рассылается всем узлам сети с данным номером сети. Такая рассылка называется *широковещательным сообщением (Broadcast)*.
   Из этих двух пунктов видно, что в любой сети два значения номера узла зарезервированы для служебной надобности.
6) Если первый октет адреса равен 127, то адрес обозначает тот же самый узел. Такой адрес используется для взаимодействия процессов на одной и той же машине и называется *возвратным (Loopback)* адресом.

5.1.2.3. Подсети

Стандартная схема разбиения пула адресов на классы порождает ряд проблем, как то:

— резкий рост таблиц маршрутизации в Интернете;
— нерациональное использование адресного пространства.

Для решения данных проблем был введён дополнительный уровень иерархии структуры IP-адреса — номер подсети (рис. 5.5).

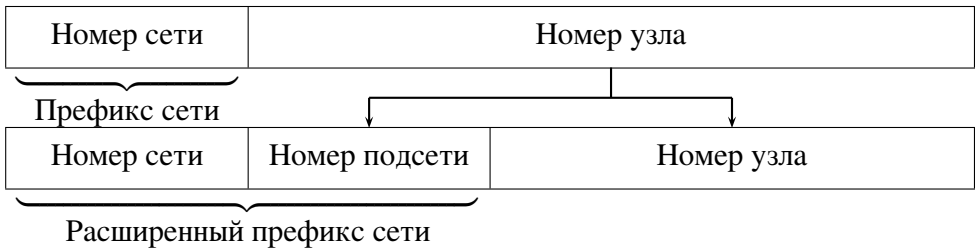


Рис. 5.5. Двухуровневая и трёхуровневая иерархии IP-адресов

Таким образом, снаружи адресация проводится по номеру сети, а внутренняя организация сети не видна извне. Любое изменение топологии внутренней сети не влияет на таблицы маршрутизации в Интернете. Это уменьшает первую проблему. С другой стороны, разбиение на подсети позволяет организации, которой выделена сеть, более гибко и экономно использовать адресное пространство, что смягчает вторую проблему.

5.1.2.4. Маска подсети

Поля номеров сети и подсети образуют *расширенный сетевой префикс*. Для выделения расширенного сетевого префикса используется *маска подсети (Subnet Mask)* — 32-разрядное двоичное число (по длине IP-адреса), в разрядах расширенного префикса содержащая единицу, а в остальных разрядах — ноль. Расширенный сетевой префикс получается побитным сложением по модулю два (операция XOR) IP-адреса и маски подсети. При таком построении очевидно, что число подсетей представляет собой степень двойки — 2^n, где n — длина поля номера подсети. Таким образом, характеристики IP-адреса полностью задаются собственно IP-адресом и маской подсети.

Для упрощения записи применяют следующую нотацию (так называемая *CIDR-нотация*): `IP-адрес/длина_ расширенного_сетевого_префикса`. Например, адрес 192.168.0.1 с маской 255.255.255.0 будет в данной нотации выглядеть как

192.168.0.1/24 (очевидно, что 24 — это число единиц, содержащихся в маске подсети).

Для стандартных классов сетей можно записать следующие значения масок подсетей (в десятично-точечной нотации):

— 255.0.0.0 — маска для сети класса A; длина расширенного сетевого префикса — 8;
— 255.255.0.0 — маска для сети класса B; длина расширенного сетевого префикса — 16;
— 255.255.255.0 — маска для сети класса C; длина расширенного сетевого префикса — 24.

5.1.2.5. Маска подсети переменной длины

В RFC 1009 [11] был определён порядок использования в сети, разделённой на подсети, нескольких масок подсети. В этом случае расширенные сетевые префиксы имеют разную длину, и маски подсетей называются *масками подсетей переменной длины (Variable Length Subnet Mask, VLSM)*. Таким образом можно разбить сеть на подсети разного размера.

Подсети выделяются рекурсивно: сеть разбивается на подсети, далее некоторые из этих подсетей в свою очередь тоже делятся на подсети и т.д. При этом, количество доступных подсетей вычисляется по формуле 2^m, где m — количество бит, выделяемых для идентификации сети, а количество узлов в сети вычисляется по формуле $2^n - 2$, где n — количество бит, выделяемых для идентификации узла.

В качестве примера на рис. 5.6 приведено разбиение сети класса C.

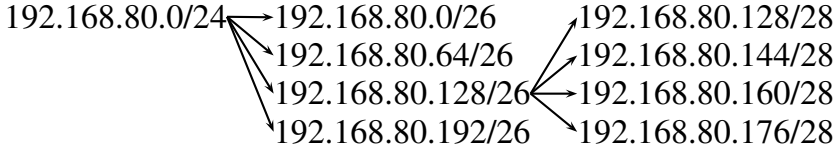


Рис. 5.6. Разбиение сети на подсети

Рассмотрим подробнее разбиение сети на несколько подсетей.

Пример (Разбиение сети класса A на 5 подсетей).

Пусть дана сеть 100.0.0.0/8. Длина сетевого префикса — 8. Сети соответствует маска 255.0.0.0 и broadcast-адрес 100.255.255.255/8. Данную сеть можно разбить на $2^8 = 256$ подсетей.

Разобьём сеть сначала на 4 подсети. Для этого под идентификатор сети надо выделить дополнительно 2 бита, чтобы получить 4 различные комбинации: `00`, `01`, `10`, `11`. Эти комбинации бит определяют вид второго октета адреса: `00000000` даёт число 0 в десятичной форме записи, `01000000` — число 64, `10000000` — число 128, `11000000` — число 192.

Сетевой префикс будет иметь длину 10, а маска будет иметь вид 255.192.0.0 (или в двоичной форме `11111111 11000000 00000000 00000000`).

Чтобы получить broadcast-адрес, надо зафиксировать сетевую часть адреса, а биты, относящиеся к номеру хоста, положить равными 1. Так сеть 100.64.0.0/10 в двоичной форме запишется как

```
01100100 01000000 00000000 00000000.
```

Её broadcast-адрес в двоичной форме будет иметь вид

```
01100100 01111111 11111111 11111111,
```

а в десятичной форме — 100.127.255.255.

Таким образом, получим 4 подсети.

адрес подсети	broadcast-адрес	маска
100.0.0.0/10	100.63.255.255/10	255.192.0.0
100.64.0.0/10	100.127.255.255/10	255.192.0.0
100.128.0.0/10	100.191.255.255/10	255.192.0.0
100.192.0.0/10	100.255.255.255/10	255.192.0.0

Одну из подсетей, например 100.64.0.0/10, разобьём ещё на 2 подсети. Для этого под идентификатор сети надо выделить дополнительно ещё 1 бит. Маска будет иметь вид 255.224.0.0. Тогда получим ещё 2 подсети:

адрес подсети	broadcast-адрес	маска
100.64.0.0/11	100.81.255.255/11	255.224.0.0
100.96.0.0/11	100.127.255.255/11	255.224.0.0

5.1.2.6. Бесклассовая маршрутизация

Логическим продолжением концепции подсетей является концепция *бесклассовой маршрутизации (Classless InterDomain Routing, CIDR)*, RFC 1517–1520 [12, 13, 14, 15]. Основные положения данной технологии:

— Отход от традиционной концепции разделения IP-адресов на классы. Вместо этого для определения границ между номером сети и номером хоста используется расширенный сетевой префикс. Таким образом, возможна организация сетей произвольного размера. При маршрутизации каждая часть маршрутной информации распространяется маршрутизаторами совместно с сетевым префиксом.
— Объединение маршрутов. Рекомендуется выделять адреса иерархически.

5.2. Протокол IPv6

Для решения проблемы нехватки адресного пространства был разработан протокол IPv6 со 128-битными IP-адресами. Кроме того, IPv6 призван решить и некоторые другие задачи (улучшение масштабируемости сети, поддержка качества обслуживания, обеспечение информационной безопасности и др.).

По сравнению с IPv4 в IPv6 внесены следующие изменения:

— упрощён заголовок IP-пакета;
— расширено адресное пространство с 32 до 128 бит;
— улучшена поддержка иерархической адресации, агрегирования маршрутов и автоматического конфигурирования адресов;
— добавлены механизмы аутентификации и шифрования на уровне IP-пакетов;
— добавлены метки потоков данных.

5.2.1. Формат заголовка пакета IPv6

IP-пакет состоит из заголовка и поля данных. Формат заголовка IPv6 определён в RFC-2460 [16] (ранее — RFC-1883 [17]). Длина заголовка — 40 байт. Рассмотрим его структуру (рис. 5.7).

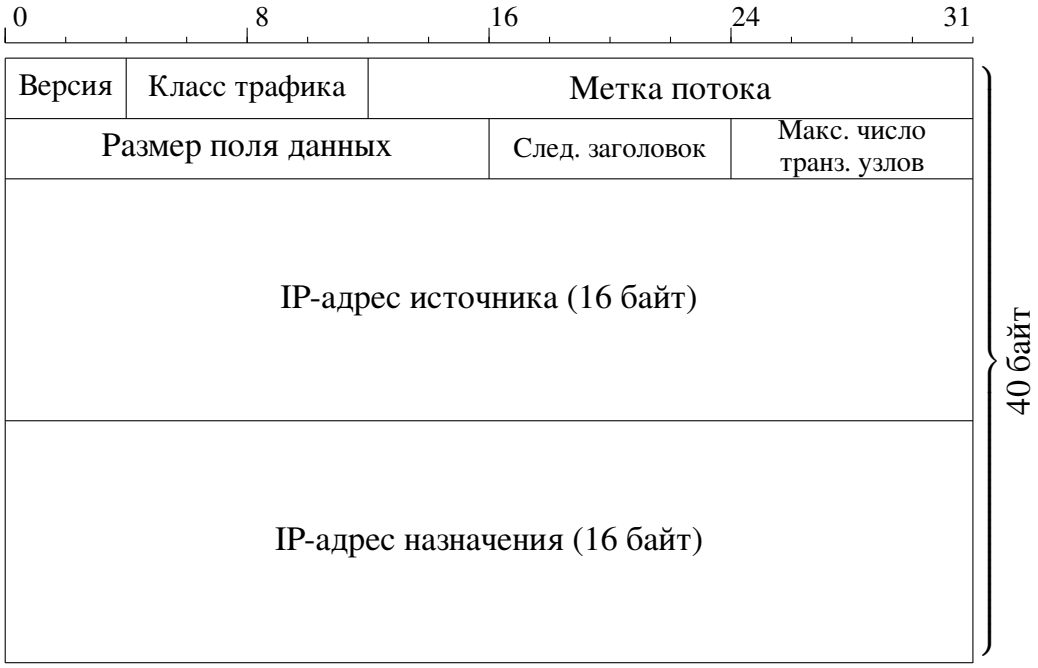


Рис. 5.7. Формат заголовка пакета IPv6 (RFC-2460)

Поле *Версия (Version)* (длина 4 бита) указывает версию протокола IP. В данном случае IPv6.

Поле *Класс трафика (Traffic Class)* (длина 8 бит) предназначено для определения класса и соответствующего приоритета пакета. В RFC-1883 данное поле называлось *Приоритет (Priority)*, имело длину 4 бита и указывало приоритет пакета, причём 16 значений этого поля были разделены на две категории: значения поля от 0 до 7 применялись к пакетам, от которых можно отказаться при перегрузке линии; значения поля от 8 до 15 назначались пакетам, которые должны быть отправлены при любом состоянии (кроме обрыва) линии.

Поле *Метка потока (Flow Label)* (длина 20 бит) идентифицирует поток, требующий специальной обработки (например, определённой полосы пропускания или задержки) сетевыми модулями. Меткой потока служит псевдо-случайное число, которое также может служить хеш-ключом для шлюзов, обрабатывающих определённый поток. В RFC-1883 данное поле имело длину 24 бита.

Поле *Размер поля данных (Payload Length)* (длина 16 бит) указывает длину (в октетах) поля данных, которое следует сразу после заголовка пакета.

Поле *Следующий заголовок (Next Header)* (длина 8 бит) идентифицирует тип заголовка, который следует непосредственно за заголовком IPv6 — дополнительный заголовок IPv6, следующий за основным, или идентификатор протокола верхнего уровня, которому принадлежит информация, размещённая в поле данных пакета.

Поле *Максимальное число транзитных узлов (Hop Limit)* (длина 8 бит) указывает предельный срок, в течение которого пакет может перемещаться по сети. Величина этого поля уменьшается на 1 при прохождении пакета через узел сети (шлюз или хост). Если величина этого поля равна 0, пакет уничтожается.

Поля *IP-адрес источника (Source IP Address)* и *IP-адрес назначения (Destination IP Address)* (длина по 128 бит) содержат адреса отправителя и получателя соответственно.

5.2.2. Дополнительные заголовки IPv6

Протокол IPv6 предусматривает возможность размещения дополнительных заголовков после основного. Дополнительные заголовки представляют собой блоки данных, каждый из которых отвечает за выполнение собственных функций. В каждом из них содержится поле идентификатора следующего заголовка. В настоящее время определены следующие дополнительные заголовки: маршрутизации, фрагментации, аутентификации, опций Hop-by-Hop, места назначения и отсутствия следующего заголовка.

Заголовок *опций Hop-by-Hop* содержит информацию, которая должна проверяться на каждом узле по пути следования пакета. Его идентификатор в основном заголовке — число 0. Структура заголовка представлена на рис. 5.8.

Рис. 5.8. Структура дополнительного заголовка опций Hop-by-Hop

Заголовок содержит следующие поля:

— поле *Следующий заголовок (Next Header)* (длина 8 бит) идентифицирует тип следующего заголовка;
— поле *Длина дополнительного заголовка (Hdr Ext Len)* (длина 8 бит) указывает длину данного заголовка в 64-битных единицах (не считая первые 64 бита);
— поле *Опции (Options)* содержит параметры, определяющие некоторые стандартные операции над пакетом, а также параметр Jumbo Payload (сверхдлина), использующийся в тех пакетах, длина которых более 65535 байт, и указывающий длину пакета в байтах, включая длину заголовка опций Hop-by-Hop, и должно быть больше 65535 байт. При этом в поле *Размер поля данных (Payload Length)* основного заголовка проставляется значение 0.

Заголовок *маршрутизации* рис. 5.9 используется отправителем IPv6 для того, чтобы указать пакету список промежуточных узлов, через которые пакет должен пройти по пути к адресу назначения. Этот заголовок идентифицируется значением 43 в заголовке предыдущего заголовка и содержит следующие поля:

— поле *Следующий заголовок (Next Header)* (длина 8 бит) идентифицирует тип следующего заголовка;

— поле *Длина дополнительного заголовка (Hdr Ext Len)* (длина 8 бит) указывает длину данного заголовка в 64-битных единицах (не считая первые 64 бита);
— поле *Тип маршрутизации (Routing Type)* (длина 8 бит) содержит идентификатор конкретного варианта маршрутизации;
— поле *Оставшиеся сегменты (Segment Left)* (длина 8 бит) содержит число промежуточных узлов, которые должны быть посещены пакетом по пути к месту назначения;
— поле *Специальный тип данных (Type-Specific Data)* (длина кратна 8 октетам) имеет формат, зависящий от кода поля *Тип маршрутизации (Routing Type)*, длина определяется заголовком маршрутизации.

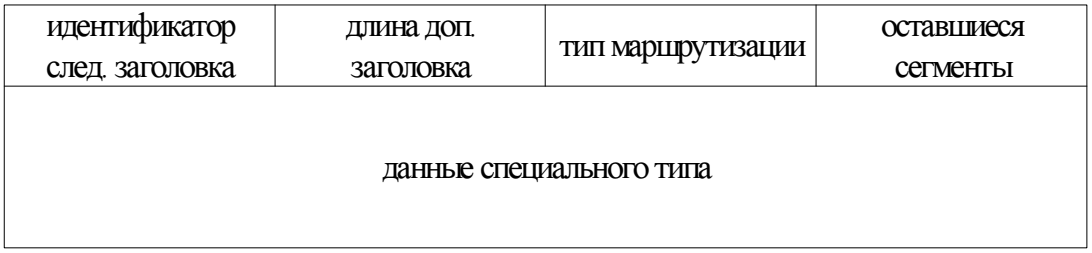


Рис. 5.9. Структура дополнительного заголовка маршрутизации

Заголовок *фрагментации* используется отправителем IPv6 для посылки пакетов длиннее, чем MTU пути до места назначения. Заголовок фрагментации идентифицируется значением 44 в заголовке предыдущего заголовка. Структура заголовка представлена на рис. 5.10.

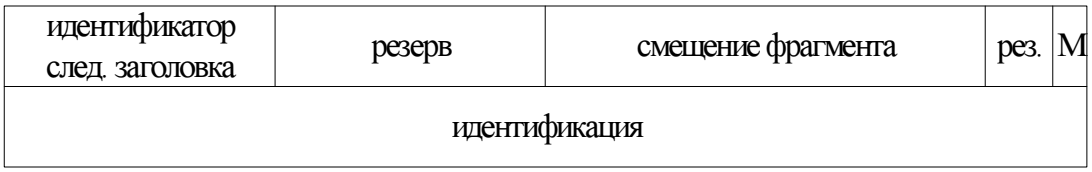


Рис. 5.10. Структура дополнительного заголовка фрагментации

Заголовок содержит следующие поля:
— поле *Следующий заголовок (Next Header)* (длина 8 бит) идентифицирует тип следующего заголовка;
— поле *Резерв (Reserved)* (длина 8 бит) инициализируется нулем при передаче и игнорируется при приёме;
— поле *Смещение фрагмента (Fragment Offset)* (длина 13 бит) задаётся в 64-битных единицах по отношению к началу фрагментируемой части пакета;
— поле *Резерв (Res)* (длина 2 бита) принимает значение 0 в случае передачи и игнорируется в случае приёма пакета;
— поле *Флаг M* (длина 1 бит) принимает значение 1, если имеется следующий фрагмент, и 0 — если этот фрагмент последний;
— поле *Идентификатор (Identification)* (длина 32 бита) содержит идентификатор фрагмента.

Заголовок *аутентификации* представляет собой механизм, который обеспечивает целостность передаваемых данных и аутентификацию отправителя на уровне IP-протокола как для IPv4, так и для IPv6. Он обеспечивает защиту передаваемой информации благодаря шифрованию данных на основе криптографического ключа с использованием асимметричных методов кодирования (например, RSA). Данный механизм обеспечивает целостность данных добавлением к IP-пакету информации аутентификации. Эта информация определяется на основе содержимого всех полей IP-пакета (как заголовков, так и пользовательских данных), которые не изменяются при передаче пакета. Заголовок аутентификации идентифицируется значением 51 в заголовке предыдущего заголовка. Содержит следующие поля:

— поле *Следующий заголовок (Next Header)* (длина 8 бит) идентифицирует тип следующего заголовка;
— поле *Длина (Length)* (длина 8 бит) содержит длину данных аутентификации в 32-битных единицах;
— поле *Резерв (Reserved)* (длина 16 бит) не используется;
— поле *Индекс параметров безопасности (Security Parameters Index)* (длина 32 бита) представляет собой псевдослучайное число, определяющее индекс соответствия в SA (определяющей тип алгоритма, параметры шифрования и т.д.);
— поле *Данные аутентификации (Authentication Data)* содержит результат работы алгоритма шифрования.

Заголовок *места назначения* используется для передачи опционной информации, которая должна анализироваться только узлом (узлами) назначения. Заголовок опции места назначения идентифицируется значением 60 предшествующего заголовка. Структура заголовка представлена на рис. 5.11.

Рис. 5.11. Структура дополнительного заголовка места назначения

Содержит следующие поля:

— поле *Следующий заголовок (Next Header)* (длина 8 бит) идентифицирует тип следующего заголовка;
— поле *Длина дополнительного заголовка (Hdr Ext Len)* (длина 8 бит) указывает длину данного заголовка в 64-битных единицах (не считая первые 64 бита);
— поле *Опции (Options)* содержит параметры, определяющие некоторые операции над пакетом.

Заголовок *отсутствия следующего заголовка* указывает, что за этим заголовком ничего не следует. Заголовок отсутствия следующего заголовка идентифицируется значением 59 предшествующего заголовка.

5.2.3. Схема адресации протокола IPv6

Схема адресации протокола IPv6 описана в документах RFC 3513 [18] (ранее RFC-2373, RFC-1884), RFC 3531 [19], RFC 3587 [20].

5.2.3.1. Форма записи адреса

Адрес IPv6 имеет длину 128 бит, разделяется на части по 16 бит, которые преобразуются в 4-значные шестнадцатеричные числа и разделяются двоеточиями. Получающаяся форма записи называется *двухточечно-шестнадцатеричной*.

Ниже показан адрес IPv6 в двоичной форме:

```
0010000111011010 0000000011010011 0000000000000000
0010111100111011 0000001010101010 0000000011111111
1111111000101000 1001110001011010
```

и соответствующая его запись в двухточечно-шестнадцатеричной форме:

```
21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A
```

Представление адресов IPv6 может быть упрощено путём удаления начальных нулей в каждом 16-битном блоке. Однако каждый из блоков должен содержать не менее одного знака. При подавлении начальных нулей адрес выглядит следующим образом:

```
21DA:D3:0:2F3B:2AA:FF:FE28:9C5A
```

Некоторые типы адресов содержат длинные последовательности нулей. Для дальнейшего упрощения адресов IPv6 сплошные последовательности 16-битных блоков из нулей в двухточечно-шестнадцатеричном формате могут быть сокращены до :: (т.н. двойное двоеточие). Например, адрес

```
FE80:0:0:0:2AA:FF:FE9A:4CA2
```

можно сократить до

```
FE80::2AA:FF:FE9A:4CA2,
```

а адрес

```
FF02:0:0:0:0:0:0:2
```

можно сократить до

```
FF02::2.
```

Сокращение нулей можно использовать только для сокращения единого сплошного ряда 16-битных блоков в двухточечно-шестнадцатеричном формате и только один раз. В противном случае будет невозможно определить число нулей, представленных каждым двойным двоеточием.

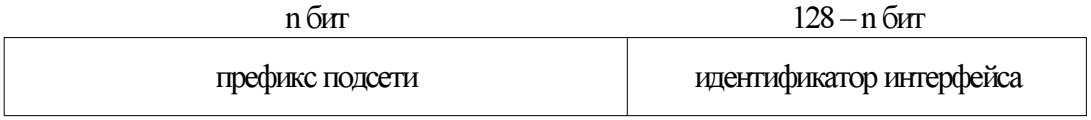


Рис. 5.12. Префикс в структуре адреса IPv6

5.2.3.2. Понятие префикса

Аналогично способу разделения пространства адресов IPv4, пространство адресов IPv6 делится согласно значениям старших битов адреса, которые образуют *префикс формата (Format Prefix, FP)* (рис. 5.12).

Префикс — это часть адреса, указывающая биты, имеющие фиксированные значения, или биты, идентифицирующие сеть. Префиксы для маршрутов IPv6 и идентификаторов подсети выражаются так же, как в нотации CIDR для IPv4.

Префикс IPv6 записывается в нотации `адрес/длина_префикса`. Например, `21DA:D3::/48` — префикс маршрута, а `21DA:D3:0:2F3B::/64` — префикс подсети.

5.2.3.3. Типы адресов

Адреса IPv6 всех типов ассоциируются с интерфейсами, а не с узлами. Интерфейс принадлежит только одному узлу, узел может иметь несколько интерфейсов. Одному интерфейсу могут соответствовать много адресов IPv6 различного типа.

Существует три типа адресов:

— *адрес одиночного интерфейса (Unicast),*
— *адрес любого интерфейса группы интерфейсов (Anycast),*
— *адрес группы интерфейсов (Multicast)*.

Краткая сводка по адресам дана в табл. 5.4.

5.2.3.3.1. Адрес одиночного интерфейса (Unicast). *Адрес одиночного интерфейса (Unicast)* идентифицирует только один интерфейс, за исключением двух случаев:

1) если приложение рассматривает несколько интерфейсов как единое целое на уровне Интернета;
2) если на маршрутизаторе есть ненумерованные интерфейсы (интерфейсу не присваивается никакого IPv6 адреса) для соединений точка-точка.

Существует несколько форм адресов одиночного интерфейса:

— глобальный адрес одиночного интерфейса провайдера (Global Provider Based Unicast Address);
— адрес локальной связи (Link-Local-Use Address);
— адрес локальной подсети (Site-Local-Use Address);
— адреса совместимости с адресами IPv4;
— адрес NSAP (Network Service Access Point).

Глобальный адрес одиночного интерфейса провайдера (Global Provider Based Unicast Address) определяется префиксом `001`, эквивалентен общедоступному адресу IPv4, применяется для глобальной маршрутизации в части Интернет, использующей протокол IPv6. Структура адреса представлена на рис. 5.13 и 5.14.

Таблица 5.4

Префиксы адресов IPv6

Назначение	Префикс (двоичный)
Зарезервировано	0000 0000
Не определено	0000 0001
Зарезервировано для NSAP	0000 001
Зарезервировано для IPX	0000 010
Не определено	0000 011
Не определено	0000 1
Не определено	0001
Не определено	001
Провайдерские Unicast-адреса	010
Не определено	011
Зарезервировано для географических уникаст-адресов	100
Не определено	101
Не определено	110
Не определено	1110
Не определено	1111 0
Не определено	1111 10
Не определено	1111 110
Не определено	1111 1110 0
Адреса локальной связи (локальные канальные адреса)	1111 1110 10
Адрес локальной подсети	1111 1110 11
Адреса группы интерфейсов (Multicast-адреса)	1111 1111

Рис. 5.13. Общая структура глобального Unicast-адреса IPv6

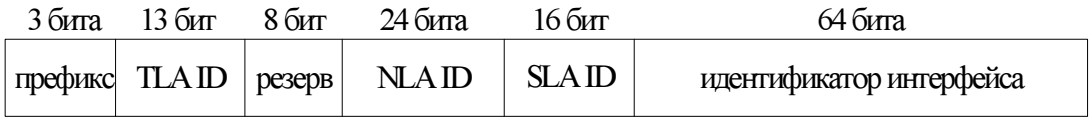


Рис. 5.14. Структура глобального Unicast-адреса провайдера

Поле *Агрегированный идентификатор верхнего уровня (TLA ID, Top Level Aggregation Identifier* (длина 13 бит)[1] указывает идентификатор составляющей верхнего уровня для адреса, т.е. самый верхний уровень в иерархии маршрутизации. Распределением адресов данного диапазона занимается *IANA (Internet Assigned Numbers Authority)*, предоставляя адреса локальным реестрам Интернета, которые, в свою очередь, выделяют отдельные идентификаторы TLA крупным поставщикам услуг Интернета. Маршрутизаторы на верхнем уровне иерархии маршрутизации по протоколу IPv6 не имеют маршрутов по умолчанию — только маршруты с 16-битными префиксами, соответствующими выделенным TLA.

Поле *Резерв (Res)* (длина 8 бит) зарезервировано для будущего использования при увеличении размера полей TLA ID или NLA ID.

Поле *Агрегированный идентификатор следующего уровня (NLA ID, Next Level Aggregation Identifier)* (длина 24 бита) указывает идентификатор составляющей следующего уровня для адреса, т.е. используется для указания подсети конкретного потребителя. Поле NLA ID позволяет поставщикам услуг Интернета создавать множественные уровни иерархии адресации для организации адресации и маршрутизации, а также для определения подсетей. Структура сети поставщика услуг Интернета не видна для маршрутизаторов верхнего уровня иерархии маршрутизации.

Поле *Агрегированный идентификатор уровня подсети (SLA ID, Site Level Aggregation Identifier)* (длина 16 бит) указывает идентификатор составляющей уровня подсети для адреса, т.е. используется отдельными организациями для определения подсетей в пределах собственной сети. Структура сети организации не видна для маршрутизаторов верхнего уровня иерархии маршрутизации.

Поле *Идентификатор интерфейса (Interface ID)* (длина 64 бита) указывает интерфейс узла в конкретной подсети.

Адрес локальной связи (Link-Local-Use Address) определяется префиксом формата `11111110 10`, используется узлами при обмене данными с соседними узлами сети с единой связью. Адреса локальной связи эквивалентны адресам APIPA (Automatic Private IP Addressing) в IPv4 (использующим префикс `169.254.0.0/16`). Адрес локальной связи необходим для процессов изучения окружения и всегда настраивается автоматически. Структура адреса представлена на рис. 5.15.

Первые 64 бита всегда закреплены за адресом локальной связи и начинаются с префикса `FE80::/64`. Маршрутизатор IPv6 никогда не направляет трафик

[1] 13-битное поле позволяет создать 8 192 различных идентификатора TLA.

локальной связи за пределы канала связи.

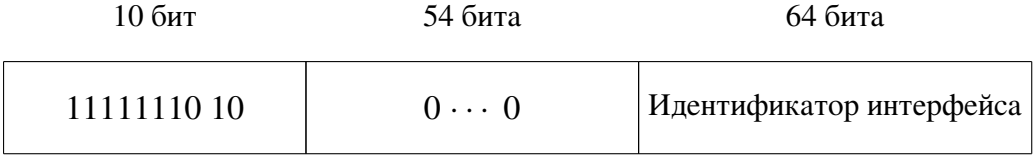


Рис. 5.15. Структура адреса локальной связи IPv6

Поле *Идентификатор интерфейса (Interface ID)* (длина 64 бита) указывает интерфейс узла.

Адрес локальной подсети (Site-Local-Use Address) определяется префиксом формата `11111110 11`. Данный тип адреса эквивалентен частному пространству адресов IPv4 (`10.0.0.0/8`, `172.16.0.0/12` и `192.168.0.0/16`). Например, частные интрасети, не имеющие непосредственного маршрутизированного подключения к Интернету на базе протокола IPv6, могут использовать адреса локальной подсети без конфликтов с глобальными адресами одиночного интерфейса провайдера. Адреса локальной подсети недоступны из других подсетей, и маршрутизаторы не должны пересылать трафик локальной подсети за пределы этой подсети. Структура адреса представлена на рис. 5.16.

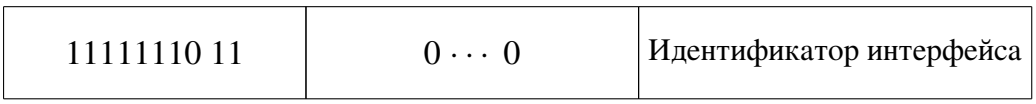


Рис. 5.16. Структура адреса локальной подсети IPv6

Первые 48 бит всегда закреплены за адресами локальной подсети и начинаются с префикса `FEC0::/48`.

Затем следует поле *Идентификатор подсети (Subnet ID)* (длина 16 бит), с помощью которого можно создавать подсети внутри организации.

Поле *Идентификатор интерфейса (Interface ID)* (длина 64 бита) указывает интерфейс узла в конкретной подсети.

Адреса совместимости с адресами IPv4 определены для облегчения перехода от адресов IPv4 к адресам IPv6:

— *IPv4-совместимый адрес* используется узлами, работающими как с протоколом IPv4, так и с протоколом IPv6. Адрес имеет вид

`0:0:0:0:0:0:w.x.y.z` или `::w.x.y.z`,

где `w.x.y.z` — точечно-десятичное представление адреса IPv4. Когда в качестве адреса назначения IPv6 используется IPv4-совместимый адрес, трафик IPv6 автоматически инкапсулируется в заголовок IPv4 и отправляется по адресу назначения с использованием инфраструктуры IPv4.

— *IPv4-сопоставленный адрес* используется для представления узла, работающего только по протоколу IPv4, узлу IPv6. Адрес имеет вид

`0:0:0:0:0:FFFF:w.x.y.z` или `::FFFF:w.x.y.z`,

где `w.x.y.z` — точечно-десятичное представление адреса IPv4. Такие адреса применяются только для внутреннего представления и никогда не используется в качестве адреса источника или назначения для пакетов IPv6.

Протокол IPv6 не поддерживает использование IPv4-сопоставленных адресов.

— *Адрес 6to4* используется узлами, работающими как с IPv4, так и с IPv6. Адрес формируется путём объединения префикса `2002::/16` с 32-битным адресом IPv4, в результате чего получается 48-битный префикс. Например, для IPv4-адреса `131.107.0.1` адрес 6to4 будет иметь вид `2002:836B:1::/48`.

Адреса NSAP (Network Service Access Point) имеют префикс формата `0000001`, а последние 121 бит адреса IPv6 сопоставляются адресу NSAP (см. RFC-1888 [21]).

5.2.3.3.2. Адрес любого интерфейса группы интерфейсов (Anycast). *Адрес любого интерфейса группы интерфейсов (Anycast)* идентифицирует некоторую группу интерфейсов. При соответствующей топологии маршрутизации пакеты, отправляемые на такой адрес, доставляются на один интерфейс (ближайший из указанных адресом). Ближайший интерфейс определяется длиной маршрута. Адрес любого интерфейса группы интерфейсов используется для обмена данными по схеме «один–один из многих» с доставкой на один интерфейс.

Для обеспечения доставки данных ближайшему члену группы инфраструктура маршрутизации должна знать интерфейсы, которым назначены такие адреса, и их удалённость в терминах метрик маршрутизации. В настоящее время этот тип адресов используются только в качестве адресов назначения маршрутизаторов и назначаются из пространства адресов одиночных интерфейсов.

Адрес любого интерфейса группы интерфейсов для маршрутизатора подсети является предопределённым и обязательным и используется для обмена данными с одним из нескольких маршрутизаторов, подключённых к удалённой подсети. Он создаётся на основе префикса данного интерфейса. При назначении такого адреса для маршрутизатора подсети значения бит префикса подсети фиксируются, а для оставшихся бит задаётся значение 0 (рис. 5.17).

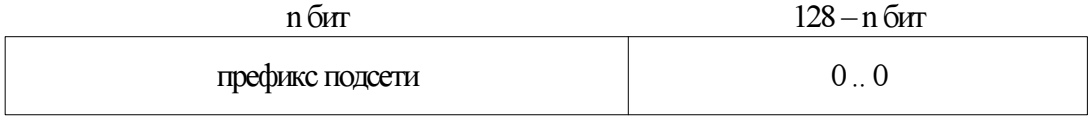


Рис. 5.17. Структура Anycast-адреса IPv6

Всем интерфейсам маршрутизатора, подключённым к подсети, назначается этот же адрес для соответствующей подсети.

5.2.3.3.3. Адрес группы интерфейсов (Multicast). При соответствующей топологии многоадресной маршрутизации пакеты, отправляемые на адрес группы интерфейсов (Multicast), доставляются на все интерфейсы, указанные адресом. Такие адреса нельзя использовать в качестве адресов источников. Структура адреса представлена на рис. 5.18.

Префикс (длина 8 бит) имеет формат `11111111`.

Поле *Флаги (Flags)* (длина 4 бита) указывает различные флаги. Согласно документу RFC-3513 [18] (ранее RFC-2373), первые три бита зарезервированы и должны быть установлены в 0, определён только последний бит — флаг T (Transient — временный). Нулевое значение флага T указывает на то, что адрес является постоянно назначенным адресом многоадресной рассылки, выделенным IANA. Значение 1 флага T указывает на то, что адрес является временным.

Рис. 5.18. Структура глобального Multicast-адреса провайдера

Поле *Область (Scope)* (длина 4 бита) указывает область объединённой сети IPv6, для которой предназначен трафик. Маршрутизаторы используют значение данного поля как дополнение к сведениям, предоставляемым протоколами многоадресной маршрутизации, для определения допустимых пределов рассылки многоадресного трафика. В документе RFC-3513 [18] (ранее RFC-2373) определены следующие значения данного поля:

- значения 0, 3, F зарезервированы;
- значение 1 — область локального интерфейса (interface-local scope);
- значение 2 — область локальной связи (link-local scope);
- значение 4 — область локального администрирования (admin-local scope);
- значение 5 — область локальной подсети (site-local scope);
- значение 8 — область локальной подсети организации (organization-local scope);
- значение E — неограниченная область (global scope);
- значения 6, 7, 9, A, B, C, D не определены.

Поле *Идентификатор группы (Group ID)* (длина 112 бит) указывает идентификатор группы многоадресной рассылки, являющийся уникальным в пределах области. Постоянно назначенные идентификаторы групп независимы от области. Временные идентификаторы групп относятся только к конкретной области. Адреса многоадресной рассылки с `FF01::` по `FF0F::` являются зарезервированными адресами.

Так, `FF01::1` указывает адрес всех узлов локального узла, `FF02::1` — адрес всех узлов локальной связи, `FF01::2` — адрес всех маршрутизаторов локального узла, `FF02::2` — адрес всех маршрутизаторов локальной связи и, наконец, `FF05::2` — адрес всех маршрутизаторов локальной подсети.

При использовании в идентификаторе группы только последних 32 бит (остальные биты этого поля должны быть установлены в 0) каждый идентификатор группы сопоставляется с уникальным MAC-адресом многоадресной рассылки Ethernet.

Адрес запроса узлов (Solicited-node Multicast Address) совмещает в себе функции адреса одиночного интерфейса и адреса Anycast. Адрес состоит из префикса `FF02::1:FF00:0/104` и последних 24 бит Unicast или Anycast-адреса IPv6. Например, узлу с адресом IPv6 локальной связи

```
FE80::2AA:FF:FE28:9C5A
```

соответствует адрес запроса узлов

```
FF02::1:FF28:9C5A.
```

5.2.3.4. Адреса серверов и маршрутизаторов

Узел IPv4 с одним сетевым адаптером обычно имеет один адрес IPv4. Узел же IPv6 обычно имеет несколько адресов IPv6, даже при наличии только одного интерфейса.

Узлу IPv6 назначаются следующие Unicast-адреса:

— адрес локальной связи для каждого интерфейса;
— адреса для каждого интерфейса (это может быть адрес локальной подсети и один или несколько глобальных адресов);
— адрес замыкания на себя (Loopback Address).

Кроме того, каждый узел прослушивает трафик на следующих адресах многоадресной рассылки:

— адрес всех узлов в области локального узла (`FF01::1`);
— адрес всех узлов в области локальной связи (`FF02::1`);
— адрес запроса узла для каждого Unicast-адреса на каждом интерфейсе;
— адреса многоадресной рассылки для групп, присоединённых к каждому интерфейсу.

Маршрутизатору IPv6 назначаются следующие Unicast-адреса:

— адрес локальной связи для каждого интерфейса;
— Unicast-адреса для каждого интерфейса (это может быть адрес локальной подсети и один или несколько глобальных адресов);
— адрес замыкания на себя (Loopback Address).

Маршрутизатору IPv6 назначаются следующие Anycast-адреса:

— Anycast-адрес для каждой подсети;
— дополнительные Anycast-адреса (необязательно).

Кроме того, каждый маршрутизатор прослушивает трафик на следующих адресах многоадресной рассылки:

— адрес всех узлов в области локального узла (`FF01::1`);
— адрес всех маршрутизаторов в области локального узла (`FF01::2`);
— адрес всех узлов в области локальной связи (`FF02::`);
— адрес всех маршрутизаторов в области локальной связи (`FF02::2`);
— адрес всех маршрутизаторов в области локальной подсети (`FF05::2`);
— адрес запроса узла для каждого Unicast-адреса на каждом интерфейсе;
— адреса групп, присоединённых к каждому интерфейсу.

5.2.3.5. Идентификаторы интерфейса IPv6

Последние 64 бита адреса IPv6 представляют собой идентификатор интерфейса, уникальный для 64-битного префикса адреса IPv6. Идентификатор интерфейса может определяться одним из следующих способов:

— согласно RFC-3513 [18] (ранее RFC-2373), все Unicast-адреса, имеющие префиксы с 001 по 111, должны также использовать 64-битный идентификатор интерфейса, образованный из адреса *EUI-64 (Extended Unique Identifier)*;
— документ RFC-3041 [22] описывает генерируемый случайным образом идентификатор интерфейса, изменяющийся со временем для обеспечения определённого уровня анонимности;
— идентификатор интерфейса, назначаемый при автоматической настройке адреса с ведением базы данных (например, по протоколу DHCPv6);
— идентификатор интерфейса, настраиваемый вручную.

5.2.3.5.1. Сопоставление адресов EUI-64 с идентификаторами интерфейсов IPv6. Адреса IEEE EUI-64 представляют новый стандарт адресации сетевых интерфейсов. Длина идентификатора компании по-прежнему составляет 24 бита, но идентификатор расширения имеет длину 40 бит, что создаёт гораздо большее пространство адресов для изготовителей сетевых адаптеров. Адрес EUI-64 использует биты U/L и I/G точно так же, как адрес IEEE 802.

Адрес EUI-64 из адреса IEEE 802 образуется путём вставки между идентификатором компании и идентификатором расширения в адресе IEEE 802 следующих 16 бит: `11111111 11111110` (`0xFF 0xFE`).

64-битный идентификатор интерфейса для Unicast-адресов IPv6 создаётся путём инвертирования бита U/L в адресе EUI-64 (значение 1 обращается в 0; значение 0 обращается в 1). Для получения идентификатора интерфейса IPv6 из адреса IEEE 802 необходимо сначала сопоставить этот адрес IEEE 802 с адресом EUI-64, а затем инвертировать бит U/L.

Пример 1. Преобразование адреса IEEE 802.

Узел имеет MAC-адрес Ethernet `00:AA:00:3F:2A:1C`. Сначала этот адрес преобразуется в формат EUI-64 путём вставки разрядов `FF:FE` между третьим и четвёртым байтами:

```
00:AA:00:FF:FE:3F:2A:1C.
```

Затем инвертируется бит U/L (седьмой бит в первом байте). Первый байт в двоичной форме имеет вид `00000000`. При инвертировании седьмого бита он принимает вид `00000010` (`0x02`). Конечный результат

```
02:AA:00:FF:FE:3F:2A:1C
```

после преобразования в двухточечно-шестнадцатеричную нотацию становится идентификатором интерфейса:

```
2AA:FF:FE3F:2A1C.
```

Таким образом, сетевому адаптеру с MAC-адресом `00:AA:00:3F:2A:1C` соответствует адрес локальной связи `FE80::2AA:FF:FE3F:2A1C`.

5.2.3.5.2. Идентификаторы интерфейса временного адреса. Пользователь Интернета при подключении к поставщику услуг Интернета получает адрес IPv4 с использованием протоколов *PPP (Point-to-Point Protocol)* и *IPCP (Internet Protocol Control Protocol)*. Адрес IPv4 может изменяться при каждом подключении данного пользователя. В связи с этим сложно отследить трафик пользователя в Интернете по IP-адресу.

При подключении удалённого доступа на базе протокола IPv6 после установки соединения пользователю назначается 64-битный префикс (путём обнаружения маршрутизатора и автонастройки адреса). Если идентификатор интерфейса всегда основан на адресе EUI-64 (полученном из статического адреса IEEE 802), то существует возможность определения трафика конкретного узла независимо от его префикса, что облегчает отслеживание конкретных пользователей. Для обеспечения определённого уровня анонимности в документе RFC-3041 [22] описан альтернативный идентификатор интерфейса IPv6, генерируемый случайным образом и изменяемый с течением времени.

Исходный идентификатор интерфейса генерируется с использованием случайных чисел. Для систем IPv6, не способных хранить протокольные сведения для создания будущих значений идентификатора интерфейса, при каждой инициализации протокола IPv6 генерируется новый случайный идентификатор интерфейса. Для систем IPv6, имеющих возможность хранения протокольных сведений, при инициализации протокола IPv6 новый идентификатор интерфейса создаётся следующим образом:

1) загружается значение из хранилища протокольных сведений и добавляется к идентификатору интерфейса, основанному на адресе EUI-64 адаптера;
2) для величины, полученной на шаге 1, вычисляется хеш-функция MD5;
3) последние 64 бита хеша MD5, вычисленного на шаге 2, сохраняются для вычисления следующего идентификатора интерфейса;
4) для седьмого из первых 64 бит хеша MD5, вычисленного на шаге 2, задаётся значение 0 (седьмым является бит U/L, нулевое значение которого задаёт локальное администрирование идентификатора интерфейса), результат которого представляет собой идентификатор интерфейса.

Итоговый адрес IPv6, основанный на таком случайном идентификаторе интерфейса, называют *временным адресом*. Временные адреса создаются для префиксов общедоступных адресов, использующих автонастройку адресов без ведения базы данных.

5.3. Другие протоколы межсетевого уровня стека TCP/IP

5.3.1. Протокол ICMP

Протокол передачи команд и сообщений об ошибках (Internet Control Message Protocol, ICMP) используется программным обеспечением ЭВМ при взаимодействии друг с другом в рамках идеологии TCP/IP (см. RFC 792 [23]).

5.3.1.1. Функциональность протокола ICMP

ICMP-протокол осуществляет:

— передачу отклика на пакет или эхо на отклик;
— контроль времени жизни дейтаграмм в системе;
— реализует переадресацию пакета;
— выдаёт сообщения о недостижимости адресата или о некорректности параметров;
— формирует и пересылает временные метки;
— выдаёт запросы и отклики для адресных масок и другой информации.

ICMP-сообщения об ошибках никогда не выдаются в ответ на:

— ICMP-сообщение об ошибке;
— при мультикастинге или широковещательной адресации;
— для фрагмента дейтаграммы (кроме первого);
— для дейтаграмм, чей адрес отправителя является нулевым, широковещательным или мультикастинговым.

Эти правила призваны блокировать потоки дейтаграмм, посылаемых в ответ на мультикастинг или широковещательные ICMP-сообщения.

5.3.1.2. Форматы пакетов ICMP

ICMP-сообщения могут быть нескольких типов. Поэтому все ICMP-пакеты начинаются с 8-битного поля *Тип ICMP* и поля *Код* (15 значений).

Типы ICMP:

0 — эхо-ответ (ping-отклик);
3 — адресат не достижим;
4 — отключение источника при переполнении очереди;
5 — изменить маршрут;
8 — эхо-запрос (ping-запрос);
9 — объявление маршрутизатора;
10 — запрос маршрутизатора;
11 — для дейтаграмм время жизни истекло (TTL=0);
12 — проблема с параметрами дейтаграммы;
13 — запрос временной метки;
14 — временная метка-отклик;
15 — запрос информации;
16 — информационный отклик;
17 — запрос адресной маски;
18 — отклик на запрос адресной маски.

Код уточняет функцию ICMP-сообщения (например, код 1 в типе ICMP 3 указывает на недостижимость ЭВМ, а код 12 для того же типа — на недоступность ЭВМ для данного вида сервиса).

На рис. 5.19 приведён формат эхо-запроса и отклика ICMP.

Рис. 5.19. Формат эхо-запроса и отклика ICMP

Поля *Идентификатор* (16 бит) и *Номер по порядку* (16 бит) служат для того, чтобы отправитель мог связать в пары запросы и отклики.

Поле *Тип* определяет, является ли пакет запросом (Тип=8) или откликом (Тип=0).

Поле *Контрольная сумма* представляет собой 16-разрядное дополнение по модулю 1 контрольной суммы всего ICMP-сообщения, начиная с поля *Тип*.

Поле *Данные* служит для записи информации, возвращаемой отправителю. Размер данного поля не регламентирован и определяется предельным размером IP-пакета.

Сообщение «адресат не достижим» посылается в случае, если маршрутизатор не может доставить дейтаграмму по назначению. На рис. 5.20 приведён формат ICMP-сообщения «адресат не достижим».

Поле *MTU на следующем этапе* характеризует максимальную длину пакетов на очередном шаге пересылки.

Рис. 5.20. Формат ICMP-сообщения «адресат не достижим»

По полю *Internet-заголовок (включая опции) + первые 64 бита дейтаграммы* можно определить, какой адрес оказался недостижимым.

В ситуации, когда принимающая сторона не справляется с приёмом потока, отправителю может быть послано сообщение с требованием снижения нагрузки. На рис. 5.21 приведён формат ICMP-запроса снижения загрузки.

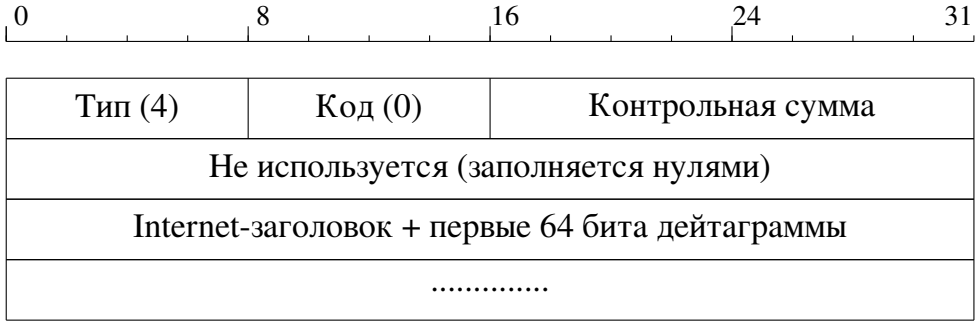


Рис. 5.21. Формат ICMP-запроса снижения загрузки

Если маршрутизатор обнаружит, что станция использует неоптимальный маршрут, он может послать ей ICMP-запрос о переадресации. Команду переадресации маршрутизатор посылает только станциям, но не маршрутизаторам. На рис. 5.22 приведён формат ICMP-запроса переадресации.

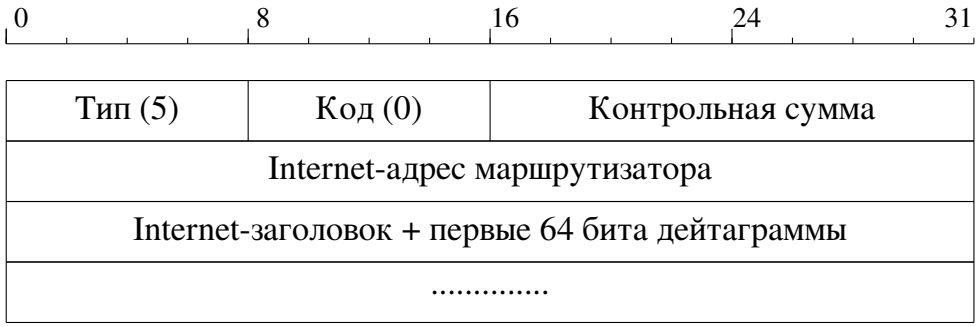


Рис. 5.22. Формат ICMP-запроса переадресации

Поле *Internet-адрес маршрутизатора* содержит адрес маршрутизатора, который станция должна использовать для отправки дейтаграммы по указанному в

заголовке месту назначения.

В поле *Internet-заголовок (включая опции) + первые 64 бита дейтаграммы* кроме самого заголовка расположены первые 64 бита дейтаграммы, вызвавшей это сообщение.

Маршрутные таблицы формируются в результате запросов и объявлений, посылаемых маршрутизаторами. Когда в сети появляется новый маршрутизатор, он посылает широковещательный запрос. В ответ другие маршрутизаторы сети посылают сообщения об имеющихся маршрутах. На рис. 5.23 приведён формат ICMP-запроса об имеющихся маршрутах.

Рис. 5.23. Формат ICMP-запроса об имеющихся маршрутах

Поле *Число адресов* указывает число адресных записей в сообщении.

Поле *Длина адреса* содержит число 32-битных слов, необходимых для описания адреса маршрутизатора.

Поле *Время жизни* указывает продолжительность жизни объявленных маршрутов в секундах.

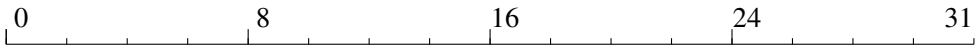
Поле *Уровень приоритета* указывает приоритет маршрута по отношению к другим маршрутам данной подсети.

На рис. 5.24 приведён формат ICMP-запроса маршрутной информации.

Рис. 5.24. Формат ICMP-запроса маршрутной информации

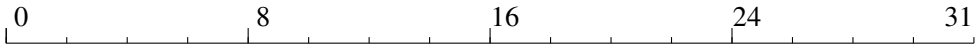
На рис. 5.25 приведён формат ICMP-запроса (отклика) маски подсети, в котором поле *Адресная маска* содержит 32-разрядную маску подсети.

При ликвидации пакета по истечении TTL маршрутизатор посылает отправителю сообщение «время истекло». На рис. 5.26 приведён формат ICMP-сообщения «время (TTL) истекло».



Тип (17 или 18)	Код (0)	Контрольная сумма
Идентификатор		Номер по порядку
Адресная маска		

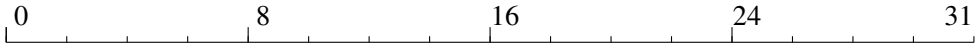
Рис. 5.25. Формат ICMP-запроса (отклика) маски подсети



Тип (11)	Код (0 или 1)	Контрольная сумма
Не используется (заполняется нулями)		
Internet-заголовок + первые 64 бита дейтаграммы		
.....		

Рис. 5.26. Формат ICMP-сообщения «время (TTL) истекло»

На рис. 5.27 приведён формат ICMP-сообщения «конфликт параметров», посылаемого маршрутизатором при выявление какой-либо ошибки (не из числа описанных выше).



Тип (12)	Код (0 или 1)	Контрольная сумма
Указатель	Не используется (заполняется нулями)	
Internet-заголовок + первые 64 бита дейтаграммы		
.....		

Рис. 5.27. Формат ICMP-сообщения типа «конфликт параметров»

Поле *Указатель* отмечает октет дейтаграммы, из-за которого возникла ошибка.

В процессе трассировки маршрутов может возникнуть проблема синхронизации времени на различных станциях. В этом случае делается запрос временно́й метки. На рис. 5.28 приведён формат ICMP-запроса временно́й метки.

Поле *Тип* со значением 13 указывает, что это запрос, а тип 14 — на то, что это отклик.

Поля *Идентификатор* (16 бит) и *Номер по порядку* (16 бит) служат для того, чтобы отправитель мог связать в пары запросы и отклики.

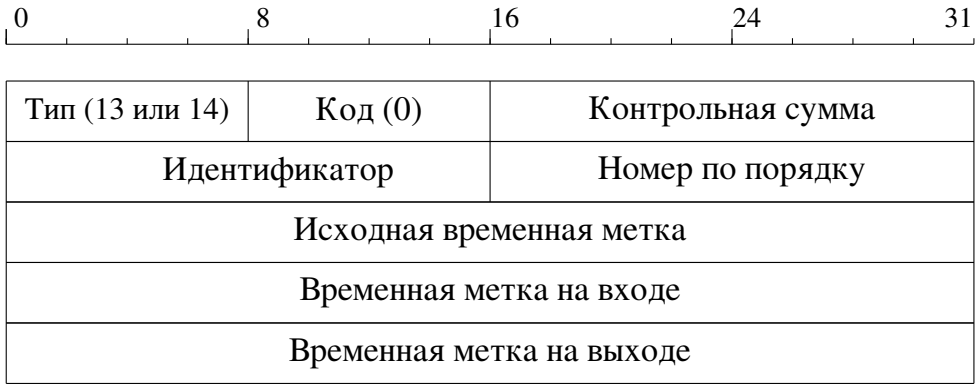


Рис. 5.28. Формат ICMP-запроса временной метки

Поле *Исходная временна́я метка* заполняется отправителем непосредственно перед отправкой пакета.

Поле *Временна́я метка на входе* заполняется маршрутизатором при получении данного пакета.

Поле *Временна́я метка на выходе* заполняется маршрутизатором непосредственно перед отправкой данного пакета.

5.3.2. Протокол ARP

Преобразование IPv4-адресов (4 байта), задаваемых с учётом положения узла в сети, в MAC-адреса (6 байт для Ethernet), заданные аппаратным образом, выполняется с помощью так называемой *ARP-таблицы* (см. RFC 826 [24]). Каждый узел сети имеет отдельную ARP-таблицу для каждого своего сетевого адаптера. Протокол *ARP (Address Resolution Protocol)* преобразует ARP-адреса в Ethernet-адреса.

5.3.2.1. Процедура преобразования адресов

При обмене сообщениями между двумя прикладными программами для определения Ethernet-адреса просматривается ARP-таблица. Если для требуемого IP-адреса в ARP-таблице присутствует Ethernet-адрес, то формируется и посылается соответствующий пакет. В противном случае выполняются следующие действия:

1) всем узлам в сети посылается пакет с ARP-запросом (с широковещательным Ethernet-адресом места назначения), а исходящий IP-пакет ставится в очередь;
2) каждый узел, принявший ARP-запрос, в своём ARP-модуле сравнивает собственный IP-адрес с IP-адресом в запросе:

   — если IP-адрес совпал, то по Ethernet-адресу отправителя запроса посылается ответ, содержащий как IP-адрес ответившего узла, так и его Ethernet-адрес, а в ARP-таблице узла-отправителя формируется соответствующий элемент и отправляется IP-пакет, ранее поставленный в очередь;

— если же в сети нет узла с искомым IP-адресом, то ARP-ответа не будет и не будет записи в ARP-таблице, а протокол IP уничтожит IP-пакеты, предназначенные этому адресу.

5.3.2.2. Формат пакета ARP

Формат пакета ARP представлен на рис. 5.29.

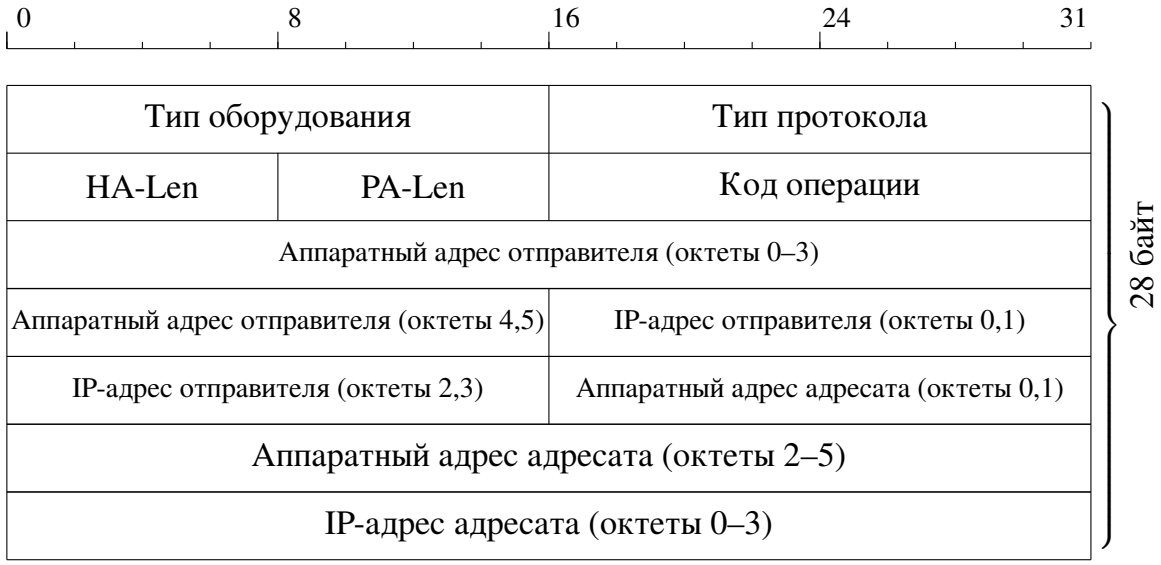


Рис. 5.29. Формат заголовка пакета ARP

Поле *Тип оборудования* (16 бит) указывает код типа интерфейса, для которого отправитель ищет адрес.

Поле *Тип протокола* (16 бит) содержит код типа протокола (например, код IP-протокола имеет значение $0800H$, код ARP-протокола — $0806H$, код RARP-протокола — $8035H$, код SNMP-протокола — $814CH$).

Поле *HA-Len* (8 бит) указывает длину аппаратного адреса.

Поле *PA-Len* (8 бит) указывает длину протокольного адреса в байтах (например, для IP-адреса *PA-Len=4*).

Поле *Код операции* (16 бит) определяет, является ли данный пакет ARP-запросом (код = 1), ARP-откликом (код = 2), RARP-запросом (код = 3) или RARP-откликом (код = 4).

Остальные поля определяют соответственно аппаратный адрес отправителя, IP-адрес отправителя, аппаратный адрес адресата, IP-адрес адресата.

5.3.3. Протокол RARP

Протокол *RARP (Reverse Address Resolution Protocol)* предназначен для обратной трансляции адресов, т.е. для преобразования MAC-адресов в IP-адреса (см. RFC 903 [25]).

Протокол RARP предполагает наличие специального сервера, обслуживающего RARP-запросы и хранящего базу данных о соответствии аппаратных адресов протокольным адресам.

5.3.3.1. Формат пакета RARP

Протокол RARP имеет сходный с ARP формат сообщения (рис. 5.30).

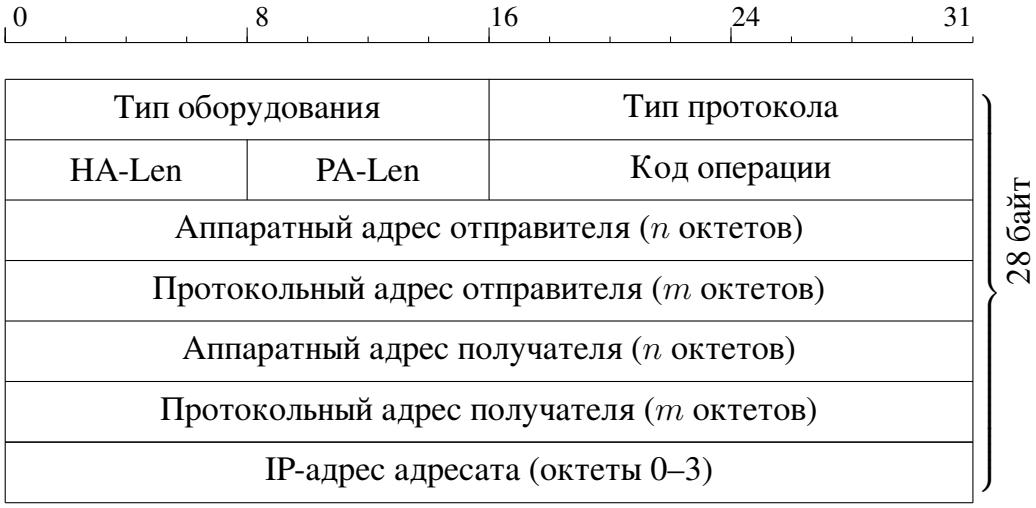


Рис. 5.30. Формат RARP-сообщения

Поле *Тип оборудования* (16 бит) указывает тип интерфейса, для которого отправитель ищет адрес (например, для Ethernet код содержит 1).

Поле *Тип протокола* (16 бит) содержит код типа протокола (например, код IP-протокола имеет значение $0800H$, код ARP-протокола — $0806H$, код RARP-протокола — $8035H$, код SNMP-протокола — $814CH$).

Поле *HA-Len* (8 бит) указывает длину аппаратного адреса (задаёт значение n).

Поле *PA-Len* (8 бит) указывает длину протокольного адреса в байтах (задаёт значение m; например, для IP-адреса *PA-Len=4*).

Поле *Код операции* (16 бит) определяет, является ли данный пакет ARP-запросом (код = 1), ARP-откликом (код = 2), RARP-запросом (код = 3) или RARP-откликом (код = 4).

5.3.3.2. Применение протокола RARP

Протокол RARP применяется, например, когда необходимо инициализировать бездисковую рабочую станцию (так как нет возможности сохранять IP-адрес на жёстком диске): для переноса из сервера в память образа операционной системы может использоваться протокол TFTP, при этом IP-адреса сервера и станции-клиента должны быть известны.

5.4. Маршрутизация

Процесс маршрутизации можно разделить на два иерархически связанных уровня:

— уровень маршрутизации,
— уровень передачи пакетов.

На уровне маршрутизации происходит работа с таблицей маршрутизации. Таблица маршрутизации служит для определения адреса (сетевого уровня) следующего маршрутизатора или непосредственно получателя по имеющемуся адресу (сетевого уровня). После определения адреса передачи выбирается определённый выходной физический порт маршрутизатора. Этот процесс называется *определением маршрута перемещения пакета*. Настройка таблицы маршрутизации осуществляется *протоколами маршрутизации (Routing Protocols)*. Примерами протоколов маршрутизации являются протоколы RIP, OSPF, BGP и др.

Уровень передачи пакетов обрабатывает команды, поступающие с уровня маршрутизации. Перед передачей пакета на этом уровне проверяется контрольная сумма заголовка пакета, определяется адрес (канального уровня) получателя пакета и производится отправка пакета с учётом очерёдности, фрагментации, фильтрации и т.д. На данном уровне используются протоколы называемые *сетевыми протоколами (Routed Protocols)*, к которым можно отнести, например, протоколы IP, IPX, AppleTalk.

Таким образом, служебная информация протоколов маршрутизации вкладывается в пакет сетевого уровня, формированием которого занимается сетевой протокол.

5.4.1. Ядерная маршрутизация

Маршрутизатор может быть реализован либо полностью *программным способом* (в этом случае он представляет собой модуль операционной системы, установленной на компьютере общего назначения, выполняющем функции сервера), либо *аппаратно-программным способом* (является специализированным вычислительным устройством, в котором часть функций выполняется нестандартной аппаратурой, а часть — программными модулями, работающими под управлением специализированной операционной системы, называемой *монитором*).

Основное преимущество программных маршрутизаторов перед аппаратными — гибкость, интеллектуальность и простота модификации алгоритмов. Возможны реализации самых нестандартных сетевых решений на базе программного маршрутизатора.

Большая часть современных программных маршрутизаторов функционирует под управлением ОС Linux, что позволяет обеспечить высокую производительность и гибкость конфигураций при осуществлении маршрутизации, а также предоставляет широкие возможности по обработке сетевого трафика, поступающего на физический интерфейс маршрутизатора.

Основная функциональность Linux-маршрутизатора обеспечивается ядром операционной системы. Любая ОС Linux, начиная с версии ядра 2.2, содержит обновлённую сетевую подсистему, архитектура которой была значительно пересмотрена и перестроена. Как результат — функциональность, превосходящая возможности аппаратно-программных маршрутизаторов, что позволяет реализовывать разнообразное управление сетевым трафиком, в частности, накладывать ограничения на транзитный трафик и осуществлять маршрутизацию на основе как идентификатора пользователя, адреса назначения, номера порта соединения, типа сервиса и других полей сетевых заголовков, так и непосредственного содержания передаваемых в пакетах данных.

В большинстве случаев настройка системы маршрутизации не представляет особых сложностей, поскольку большинство операций выполняется ядром и специализированными программными пакетами автоматически. ОС Linux силь-

на как раз своим сетевым инструментарием, поэтому данную систему можно использовать в качестве связующего звена даже в тех локальных и глобальных сетях, основную часть которых составляют компьютеры, работающие под управлением Windows, MacOS и др.

5.4.1.1. Iproute2

Ранее подсистема маршрутизации в Linux, как и большинство операционных систем UNIX, использовала утилиты arp, ifconfig и route. Но, начиная с ядра 2.2, сетевая подсистема была полностью переписана. Новый сетевой код дал увеличение производительности и более высокие эксплуатационные характеристики.

Современная реализация маршрутизации в ядре Linux основана на подсистеме iproute2. Управление на прикладном уровне представлено пакетом IProute2, входящим в большинство дистрибутивов Linux[1].

Фактически iproute2 состоит из нескольких утилит управления трафиком:

— `ip` — управление маршрутизацией;
— `tc` — управление очередями маршрутизации;
— `ss` — просмотр текущих соединений и открытых портов.

Утилита «ip» заменяет собой команды route, arp, ifconfig и предназначена для управления таблицами маршрутизации, в частности, правилами, определёнными в них, и помогает реализовывать возможности многотабличной маршрутизации, туннелирования, а также многоадресную маршрутизацию.

Кроме того, Linux имеет гибкую систему управления трафиком, называемую *Traffic Control*. Эта система поддерживает множество методов для классификации, приоритезации, разделения и ограничения как входящего трафика, так и исходящего. Управление этими функциями осуществляет вторая утилита пакета — «tc». Эта утилита также позволяет реализовать QoS в нужном для системы объёме:

— разделение разных типов трафика по классам (не только по битам ToS в IP-пакете, но и по другим данным из заголовка IP-пакета);
— назначение различных дисциплин обработки очередей трафика с разным приоритетом, механизмами прохождения очереди, ограничениями по скорости и т.п.

Обе утилиты интерпретируют выполнение команд, а все функции выполняет ядро.

Принципиальной особенностью iproute2 является использование нескольких таблиц маршрутизации. Когда ядру необходимо выбрать маршрут, оно определяет, в соответствии с какой таблицей это нужно делать. Простейшим примером использования нескольких таблиц маршрутизации является подключение сети через двух (или более) провайдеров (рис. 5.31). Такое подключение может требоваться для обеспечения отказоустойчивости или балансировки нагрузки.

Другой важной особенностью iproute2 является возможность организации туннелей. В Linux поддерживаются 3 типа туннелирования — IP-в-IP, GRE-туннелирование и туннелирование не-ядерного уровня (например, PPTP).

Туннели IP-в-IP являются самыми простыми, однако имеют ряд ограничений. Например, их организация возможна только в сетях на базе протокола IPv4. Кроме того, отсутствует интероперабельность с другими операционными системами.

[1] При этом старые команды (ifconfig, route) используют новую подсистему с некоторыми параметрами по-умолчанию.

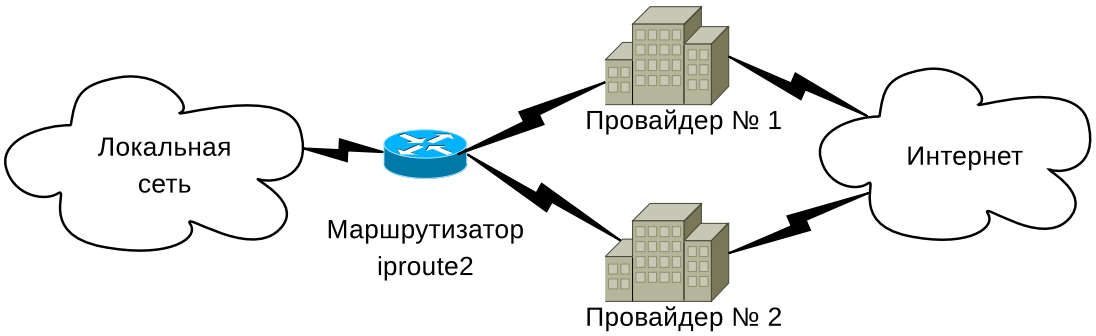


Рис. 5.31. Подключение через двух провайдеров

При этом туннели IP-в-IP могут широко применяться, например, при виртуализации.

GRE — стандартный протокол туннелирования, разработанный фирмой Cisco. В отличие от туннелей IP-в-IP, он поддерживает широковещательные сообщения и может работать в сетях на базе протокола IPv6.

Кроме маршрутизации iproute2 может осуществлять управление обработкой пакетов и, в частности, дисциплинами обработки очередей (как бесклассовыми, так и на основе классов). Также iproute2 может проводить маркировку пакетов (т.е. выполнять роль классификатора).

Несмотря на все преимущества, iproute2 имеет несколько недостатков:

1) неполная документированность;
2) маршрутизатор конфигурируется интерактивно, т.е. путём ввода команд с клавиатуры;
3) из-за своей монолитной архитектуры iproute2 имеет более высокую сложность, чем, например, Click.

5.4.1.2. Click

Click Modular Router [26] представляет собой специализированное программное обеспечение для создания высокопроизводительных программных маршрутизаторов. Click был разработан в Массачусетском технологическом университете США при поддержке национального агентства DARPA.

Маршрутизатор Click имеет модульную структуру. Отдельные элементы осуществляют простые функции маршрутизатора, такие, как классификация пакетов, организация очередей, планирование, установление связи с сетевыми устройствами. Каждый элемент представлен C++ объектом.

Конфигурация программного маршрутизатора может быть представлена в виде направленного графа с различными элементами в качестве вершин (рис. 5.32), в котором пакеты с данными перемещаются вдоль рёбер.

Благодаря архитектуре программного средства и декларативному языку описания конфигурации маршрутизатор является модульным и легко расширяемым. Высокая скорость обработки трафика достигается посредством механизма *Device Polling* (технологии работы ядра Linux с устройствами), а также за счёт внутреннего механизма аннотаций.

Как было сказано, Click состоит из набора модулей с единой системой конфигурации. Каждый модуль реализует простые функции (классификацию пакетов,

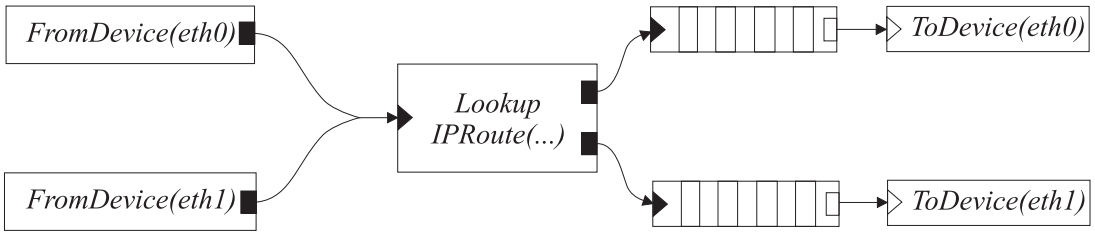


Рис. 5.32. Конфигурационный граф стандартного маршрутизатора

управление очередями, функции планировщика и интерфейса с сетевыми устройствами) и представляет собой отдельную часть процесса маршрутизации. Модуль может выполнять как простые вычисления (например, уменьшение счётчика жизни IP-пакета), так и более сложные (построение маршрута следования пакета).

Для построения маршрутизатора выбирается набор обработчиков, которые соединяются в ориентированный граф. Обработчик Click представляет собой некий элемент, в котором происходит обработка пакета. Click позволяет создавать новые обработчики, а также модифицировать и комбинировать между собой имеющиеся. Действия маршрутизатора задаются при помощи набора определённых модулей, а также путём определения связей между ними.

Для каждого элемента маршрутизатора должны быть определены:

— *класс (Element Class)*, задающий действия элемента маршрутизатора при приёме пакета;
— *порты (Ports)* для создания соединения между элементами;
— *конфигурационная строка (Configuration String)*, определяющая состояние обработчика при первом запуске программного комплекса;
— *интерфейсы (Method Interfaces)*, необходимые для обмена информацией, являющейся результатом действий элемента маршрутизатора.

Модули Click не имеют встроенных буферов для построения очередей на входных и выходных портах. Вместо этого очереди в Click реализуются специальными Queue-обработчиками. Реализация механизма построения очередей даёт возможность прямого контроля над параметрами маршрутизатора, а также возможность создания конфигураций, которые сложно реализовать другими методами.

5.4.2. Протоколы маршрутизации

Существует два основных способа определения маршрута и построения таблиц маршрутизации — *статический* и *динамический*.

5.4.2.1. Статическая маршрутизация

При использовании статического способа таблицы маршрутизации строятся администратором сети вручную. Для их построения используются специальные команды маршрутизатора (обычно это команда `route`, с помощью которой определяется маршрут для указанной сети). Параметрами этих команд служат адрес и маска сети назначения, адрес следующего маршрутизатора (next hop) для этой сети и имя или адрес интерфейса, через который должна быть передана дейтаграмма. Для корректной доставки дейтаграммы достаточно первых трёх параметров. При использовании внеклассовых сетей в таблице маршрутизации вполне могут

появиться конфликтующие маршруты, поэтому указание маски является обязательным.

Построение полной таблицы маршрутизации, в которой были бы указаны все сети, образующие Интернет, невозможно из-за огромного количества этих сетей. Для того чтобы упростить процедуру построения таблицы маршрутизации, в неё может быть включён специальный узел, куда необходимо передать дейтаграммы, адрес сети назначения которых не указан в таблице маршрутизации. Этот специальный узел называется *шлюзом по-умолчанию (Default Gateway)* и применяется для маршрутизации в режиме «по-умолчанию». Для обозначения маршрута к Default Gateway в качестве адресов сети и маски принято использовать нулевые значения.

Основным недостатком статического метода является не размер и количество создаваемых вручную таблиц маршрутизации, а тот факт, что эти таблицы фиксированные и, следовательно, не могут реально соответствовать текущей конфигурации сети (нет возможности получения информации о новых сетях и нет выбора наиболее эффективного маршрута в сети).

5.4.2.2. Динамическая маршрутизация

При использовании динамической маршрутизации формирование маршрутных таблиц производится маршрутизаторами автоматически в результате постоянного выполнения специального алгоритма маршрутизации. В процессе его выполнения маршрутизатор передаёт своим соседям информацию об известных ему маршрутах, получая от них взамен аналогичную информацию. После обработки полученной информации маршрутизатор строит заново или корректирует свою таблицу маршрутизации. Поскольку информация о состоянии маршрутов поступает на маршрутизатор постоянно, использование такого алгоритма обеспечивает постоянное соответствие содержимого таблицы маршрутизации реальному состоянию сети.

В зависимости от того, каким образом производится обмен маршрутной информацией между соседними маршрутизаторами, различают два типа алгоритмов маршрутизации:

— *алгоритмы вектора расстояния (Distance-Vector)*
маршрутизатор через заранее определённые промежутки времени передаёт соседним маршрутизаторам содержимое своей таблицы маршрутизации;
— *алгоритмы состояния канала (Link-State)*
маршрутизатор передаёт информацию только об изменениях состояния системы.

Во время построения маршрутной таблицы могут быть сформированы несколько маршрутов, ведущих в одну сеть. Для того чтобы маршрутизатор мог выбрать один из них в качестве предпочтительного, он должен использовать обобщённую характеристику качеств маршрута — *метрику (Metric)*.

Каждый алгоритм маршрутизации применяет свой алгоритм расчёта метрики. В наиболее простом случае в качестве метрики маршрута используется число узлов, отделяющих это маршрутизатор от сети назначения. Более сложные метрики учитывают характеристики физических каналов, составляющих маршрут. Некоторые алгоритмы маршрутизации для увеличения скорости информационного обмена позволяют одновременно использовать несколько маршрутов, ведущих к одной сети.

Совокупность сетей, находящихся под единым административным управлением, принято называть *автономной системой*.

Для определения внутренних маршрутов в автономных системах обычно используется один или несколько протоколов маршрутизации. В автономных системах этот класс протоколов принято называть *протоколами внутренней маршрутизации (Interior Gateway Protocol, IGP)*. Применение *протоколов внешней маршрутизации (Exterior Gateway Protocol, EGP)* позволяет администратору реализовать совокупность мер повышения надёжности и экономической эффективности информационного взаимодействия с внешними системами. В число параметров, используемых современным протоколом внешней маршрутизации для определения качества маршрута, входят предпочтительность маршрута, последовательность проходимых автономных систем и другие параметры.

5.4.2.3. Протокол RIP

Внутренний протокол маршрутизации *RIP (Routing Information Protocol)* использует алгоритм вектора расстояния для определения маршрута следования пакетов.

5.4.2.3.1. Функционирование маршрутизаторов по алгоритму вектора расстояния.

1) Маршрутизатор строит первичную таблицу маршрутизации, в которую помещает номера непосредственно подключённых сетей. Эта таблица содержит следующие поля:

   Address (Адрес) — адрес сети или узла назначения;

   Router (Маршрутизатор) — сетевой адрес первого маршрутизатора на маршруте к сети или узлу назначения;

   Interface (Интерфейс) — сетевой адрес или номер интерфейса связи с первым маршрутизатором;

   Metric (Метрика) — числовая характеристика маршрута от 0 до 15 (значение 0 соответствует непосредственно подключённой сети, метрика 15 указывает на недостижимость сети или узла назначения, в остальных случаях — соответствует количеству промежуточных маршрутизаторов на маршруте к сети или узлу назначения);

   Timer (Таймер) — показатель актуальности информации о сети или узле назначения (если информация не подтверждается источником в течение установленного временного интервала, запись о маршруте удаляется из таблицы).

2) Маршрутизатор рассылает оформленную в виде специального *сообщения об обновлении (Update)* текущую версию таблицы маршрутизации соседним маршрутизаторам.
3) При приёме аналогичного сообщения от соседнего маршрутизатора выполняются следующие действия:
   1) Если сообщение содержит информацию о сети, которой нет в таблице маршрутизации, адрес этой сети заносится в таблицу со следующими значениями полей:
      — *Router (Маршрутизатор)* — адрес источника сообщения;

- *Interface (Интерфейс)* — адрес интерфейса, принявшего сообщение;
- в поле *Metric (Метрика)* заносится значение соответствующего поля исходного сообщения, увеличенное на весовой коэффициент интерфейса (обычно все весовые коэффициенты интерфейсов принимаются равными 1);
- значение поля *Timer (Таймер)* у созданной записи устанавливается равным утроенному периоду обновлений (90 с).

2) Если сообщение содержит информацию о сети, которая есть в таблице маршрутизации, выполняется сравнение содержимого полей Router существующей записи и принятого сообщения. Если источник маршрутной информации в обоих случаях был один и тот же, поле Metric существующей записи модифицируется по обычному алгоритму значением соответствующего поля принятого сообщения. Поле Timer для модифицированной записи формируется так же, как и для вновь созданной.
3) Если информацию об известной сети содержит сообщение, принятое от нового источника, маршрутизатор сравнивает содержимое полей Metric существующей записи и принятого сообщения. Если метрика существующего маршрута больше метрики нового маршрута, прежняя запись в таблице маршрутизации заменяется новой. В противном случае таблица маршрутизации никак не модифицируется.
4) В том случае, если значение поля Timer у существующей записи стало равным 0, запись удаляется из таблицы маршрутизации.

Процессы, описанные в двух последних пунктах, периодически повторяются, что позволяет динамически отслеживать изменения конфигурации сети.

В протоколе RIP в качестве предельного значения метрики маршрута используется значение 15. Сети, удалённые от данного узла на расстояние, которое превышает 15 переходов, считаются *недостижимыми (Unreachable)*.

5.4.2.3.2. Методы противодействия возникновению циклических маршрутов. Для противодействия возникновению циклических маршрутов алгоритмы маршрутизации Distance-Vector вообще и RIP в частности используют некоторые специальные методы:

- *Правило расщеплённого горизонта (Split Horizon)*.
  Информация о маршруте в некоторую сеть N, полученная от маршрутизатора , не может быть включена в регулярные обновления, отправляемые этому маршрутизатору. Использование этой процедуры позволяет гарантированно избежать появления циклических маршрутов между двумя соседними маршрутизаторами, повышает эффективность использования пропускной способности канала за счёт сокращения неинформативной составляющей сообщения об обновлении маршрутов. Однако в том случае, если циклический маршрут образован несколькими маршрутизаторами, применение этой процедуры не даст желаемого эффекта.
- *Правило отравленного обратного пути (Poison Reverse)*.
  Действует аналогично предыдущему правилу, однако, в отличие от процедуры расщеплённого горизонта, информация о маршруте в некоторую сеть N, полученная от маршрутизатора , включается в регулярные обновления, отправляемые этому маршрутизатору с метрикой 16. В результате использования этой процедуры потенциально опасные маршруты будут удалены

из таблицы маршрутизации. Но если при использовании чистой процедуры Split Horizon эти маршруты будут удалены по истечении определённого времени, то использование Poison Reverse приведёт к их мгновенному уничтожению.

— *Метод управляемых обновлений (Triggered Update)*.
Наиболее мощным средством борьбы с длинными циклическими маршрутами является использование *апериодических управляемых обновлений маршрутов (Triggered Update)*. Маршрутизатор формирует обновление при каждом изменении своей таблицы маршрутизации, не дожидаясь наступления момента передачи очередного периодического обновления. При получении такого управляемого обновления последующий маршрутизатор скорректирует свою таблицу маршрутизации, а затем, в свою очередь, сформирует своё управляемое обновление, которое направит своим соседям. Таким образом, информация об изменении конфигурации распространяется по сети немедленно. Кроме того, вследствие особого дифференциального принципа формирования таких обновлений они распространяются по сети от источника только в нужных направлениях, поскольку маршрутизатор, не изменивший свою таблицу маршрутизации при получении управляемого обновления, не сформирует вторичное обновление и заблокирует его дальнейшее распространение.

5.4.2.3.3. Режимы RIP. При реализации RIP можно выделить следующие режимы:

— *инициализация* — посылается запрос для определения всех доступных интерфейсов;
— *получение таблиц маршрутизации* от других маршрутизаторов;
— *получен запрос* — посылается либо полная таблица маршрутизации, либо проводится индивидуальная обработка;
— *получен отклик* — проводится коррекция таблицы маршрутизации;
— *регулярные коррекции* — пересылка всей или части таблицы всем соседним маршрутизаторам каждые 30 с.

5.4.2.3.4. Формат сообщения RIP. Для взаимного обмена маршрутной информацией со своими соседями маршрутизаторы протокола RIP применяют сообщения специального формата (рис. 5.33).

Для отправки этих сообщений маршрутизаторы первой версии RIP обычно использовали широковещательный адрес (Broadcast) сетевого уровня. Особенно негативно эта особенность протокола проявлялась в сетях множественного доступа (например, Ethernet), где она могла приводить к значительному снижению эффективности использования сетевых ресурсов. В версии RIPv2 применяется специально выделенный групповой адрес 224.0.0.9 или для передачи сообщения конкретному соседу — обычный одноадресный режим (Unicast).

Сообщения протокола RIP состоят из *заголовка* и следующих за ним *маршрутных записей (Route Entries, RTE)*. Обычно в сообщении протокола RIP содержится не более 25 маршрутных записей. То есть при передаче большой таблицы маршрутизатор должен использовать несколько последовательных сообщений.

Поле *Команда (Command)* может принимать следующие значения:

1 — запрос на получение частичной или полной маршрутной информации;
2 — отклик, содержащий информацию о расстояниях из маршрутной таблицы отправителя;

3 — включение режима трассировки;
4 — выключение режима трассировки;
5–6 — зарезервированы для внутренних целей SUN Microsystem.

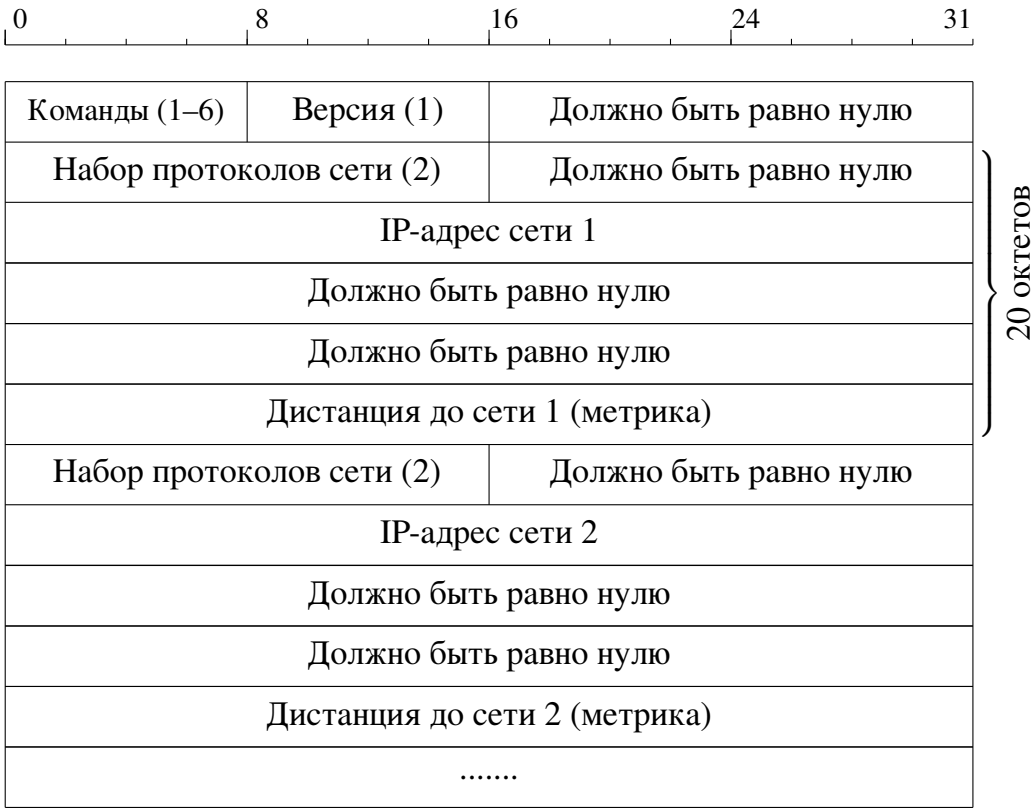


Рис. 5.33. Формат сообщения RIP

Поле *Версия (Version)* указывает версию протокола RIP (1 или 2);
Поле *Набор протоколов сети (Address family identifier)* i *(*$i \leqslant 25$*)* указывает целое число шагов до данной сети (от 1 до 15).

Сообщения типа «запрос» используются для запроса на получение полной таблицы маршрутизации или её части. Обработка запроса ведётся запись за записью (RTE за RTE). Для каждой маршрутной записи проверяется таблица маршрутизации на предмет того, есть ли там соответствующая запись. Если есть, то в поле маршрутной записи помещается метрика из таблицы маршрутизации. Если нет — в поле маршрутной записи помещается число 16, обозначающее бесконечную метрику. После того, как все маршрутные записи обработаны пакет отсылается обратно запрашивающему.

Сообщение типа «отклик» может быть ответом на конкретный запрос, регулярным сообщением обновления или сообщением обновления, вызванным изменением таблицы маршрутизации.

При получении сообщения типа «отклик» для каждого содержащегося в нём элемента вектора расстояний выполняются следующие действия:

— проверяется корректность указанных в сообщении адреса сети и маски;
— проверяется, не превышает ли метрика бесконечности:

  – некорректный элемент игнорируется;

– если метрика меньше бесконечности, она увеличивается на 1;

— в таблице маршрутов производится поиск сети, указанной в рассматриваемом элементе вектора расстояний, причём если запись о такой сети в таблице маршрутов отсутствует и метрика в полученном элементе вектора меньше бесконечности, сеть вносится в таблицу маршрутов с указанной метрикой;
— в поле «Следующий маршрутизатор» заносится адрес маршрутизатора, приславшего сообщение;
— запускается таймер для принятой записи в таблице;
— если искомая запись присутствует в таблице с метрикой больше, чем объявленная в полученном векторе, в таблицу вносятся новые записи о метрике и, соответственно, об адресе следующего маршрутизатора и таймер для этой записи перезапускается;
— если искомая запись присутствует в таблице и отправителем полученного вектора был маршрутизатор, указанный в поле «Следующий маршрутизатор» этой записи, то таймер для этой записи перезапускается;
— более того, если при этом метрика в таблице отличается от метрики в полученном векторе расстояний, в таблицу вносится значение метрики из полученного вектора;
— во всех прочих случаях рассматриваемый элемент вектора расстояний игнорируется.

5.4.2.3.5. Недостатки RIP.

— Отсутствие поддержки спецификации CIDR.
RIP-I воспринимает внеклассовые сети типа 10.1.0.0/16, 10.2.0.0/16 и т.д. как одну сеть класса A 10.0.0.0/8 и формирует для неё один маршрут, что, естественно, приводит к потере пакетов, направляемых в указанные подсети. Этот недостаток был устранён во второй версии протокола путём введения в маршрутную информацию дополнительной характеристики *SUBNET MASK (маска сети назначения)*.
— Требует много времени для восстановления связи после сбоя в маршрутизаторе.
— Возможно возникновение циклов.
— Наличие лишь одного параметра определения маршрута — числа промежуточных маршрутизаторов.

5.4.2.3.6. Протокол RIPv2. RIPv2 является расширением протокола RIPv1. Он не внёс в протокол RIPv1 каких-либо серьёзных изменений в механизме или формате сообщения, а лишь добавил возможность передачи дополнительной информации. Изменения формата заголовка пакета RIPv2 коснулись лишь поля *Версия* и ранее неиспользуемых полей, содержащих теперь дополнительную информацию.

Так в новой версии протокола появилась возможность аутентификации передаваемых сообщений, для чего используется первая маршрутная запись в заголовке пакета[1].

[1] RFC 2453 специфицирует использование только одной схемы аутентификации — использование простого нешифруемого пароля.

Кроме того, стало возможным различать «внутренние» маршруты (полученные через RIP) от «внешних» (полученных от других протоколов маршрутизации, таких, как EGP, BGP).

Как было сказано ранее, в новой версии протокола RIP стало возможным при помощи поля *Маска подсети* различать не только сети, но и подсети.

В целях уменьшения использования полосы пропускания сетей протокол RIPv2 вместо адреса broadcast использует multicast-адрес — 224.0.0.9.

5.4.2.4. Протокол OSPF

Протокол *OSPF (Open Shortest Path First)* относится к протоколам маршрутизации на основе состояния канала (класс Link-State).

5.4.2.4.1. Функционирование маршрутизаторов по алгоритму состояния каналов. Как и все протоколы маршрутизации класса Link-State, протокол OSPF предназначен для построения внутренних маршрутов *автономной системы (Autonomous System)*.

Поскольку протокол OSPF обеспечивает иерархическую маршрутизацию, автономная система разбивается на независимые области по функциональному принципу. Центральная область играет роль *магистрали (Backbone)* и используется для обеспечения информационного взаимодействия между остальными (периферийными) областями.

В зависимости от того, к какой области принадлежит маршрутизатор и какие информационные потоки через него проходят, различают четыре типа маршрутизаторов протокола OSPF:

- *внутренний маршрутизатор (Internal Router, IR)*;
- *пограничный маршрутизатор области (Area Border Router)*;
- *пограничный маршрутизатор автономной системы (AS Boundary Router, ASBR)*;
- *магистральный маршрутизатор (Backbone Router, BR)*.

Все маршрутизаторы OSPF принимают участие в формировании маршрутной информации автономной системы путём передачи специальных сообщений, содержащих информацию о текущем состоянии фрагмента сети. Эти сообщения называются *объявлением состояния канала (Link State Advertisement, LSA)*. Сообщения LSA обязательно формируются при любом изменении состояния контролируемого компонента сети. Для обеспечения большей надёжности сообщения LSA могут быть сформированы и при отсутствии каких-либо изменений в сети через достаточно большие интервалы времени, например, один раз за полчаса.

Принятые сообщения образуют в каждом маршрутизаторе *базу данных состояния сети (Link State Data Base)*. При получении сообщения об изменениях в структуре сети маршрутизатор вносит соответствующие изменения в свою копию базы данных. Таким образом, в каждый момент времени все базы данных маршрутизаторов, находящихся внутри одной автономной системы, являются идентичными и адекватно отображают структуру этой системы. Для того чтобы определить маршрут, по которому должна быть передана дейтаграмма, маршрутизатор на основании своей копии базы данных строит дерево кратчайших путей, в вершине которого размещает самого себя (используя алгоритм Дейкстры). Построение кратчайших путей маршрутизатор выполняет всякий раз, когда происходит изменение состояния сети.

Существенной особенностью протокола маршрутизации OSPF является специальная процедура информационного обмена между маршрутизаторами в сетях с множественным доступом (например, Ethernet). Маршрутизаторы, подключённые к одной и той же сети, называются *соседними маршрутизаторами (Neighboring Routers)*. Маршрутизаторы протокола OSPF устанавливают и обслуживают соседские отношения, используя специальный дополнительный протокол *Hello*. С помощью этого протокола определяется состав подключённых к сети маршрутизаторов, их работоспособность и производится выбор одного из них в качестве *назначенного маршрутизатора (Designated Router, DR)*. Назначенный маршрутизатор выбирается для того, чтобы исключить возможность многократного представления информации об одной сети. Он формирует сообщения, содержащие список подключённых к сети маршрутизаторов, и передаёт содержимое текущей базы данных по запросу, полученному от одного из них. Если по каким-либо причинам назначенный маршрутизатор перестал функционировать, его функции автоматически переходят к *запасному назначенному маршрутизатору (Backup Designated Router, BDR)*, выбираемому одновременно с основным.

Для передачи маршрутной информации маршрутизаторы протокола OSPF используют различные типы обновлений о состоянии сетевых компонентов (LSA). Процесс распространения LSA в пределах автономной системы называется *затоплением (Flooding)*.

Для хранения маршрутной информации протокола OSPF маршрутизаторы используют специальные *топологические базы данных (Link-State Database)*. База данных формируется из принятых сообщений LSA и отображает текущее состояние и структуру информационных связей в рассматриваемой области маршрутизации. На основании этой базы каждый маршрутизатор строит дерево кратчайших путей, соединяющих его самого с остальными компонентами области, и собственно таблицу маршрутизации.

5.4.2.4.2. Формат сообщений протокола OSPF. Формат заголовка сообщений протокола OSPF приведён на рис. 5.34.

Рис. 5.34. Формат заголовка сообщений протокола OSPF

Поле *Версия (Version No.)* указывает версию протокола (=2).

Поле *Тип (Packet Type)* идентифицирует функцию сообщения и может принимать следующие значения:

1 — сообщение является сообщением Hellow (используется для проверки доступности маршрутизатора);
2 — сообщение является описанием базы данных;
3 — сообщение является запросом состояния канала;
4 — сообщение информирует об изменении состояния канала;
5 — сообщение является подтверждением получения сообщения о статусе канала.

Поле *Длина пакета (Packet Length)* определяет длину блока (включая заголовок) в октетах.

Поле *Идентификатор области (Area ID)* представляет собой 32-битный код, идентифицирующий область, которой принадлежит данный пакет.

Поле *Контрольная сумма (Checksum)* содержит контрольную сумму IP-пакета, включая поле *Тип идентификации*. Контрольное суммирование производится по модулю 1.

Поле *Тип идентификации (AU type)* имеет значение 0, если отсутствует контроль доступа, и 1 — в противном случае .

Формат сообщения Hellow протокола OSPF приведён на рис. 5.35.

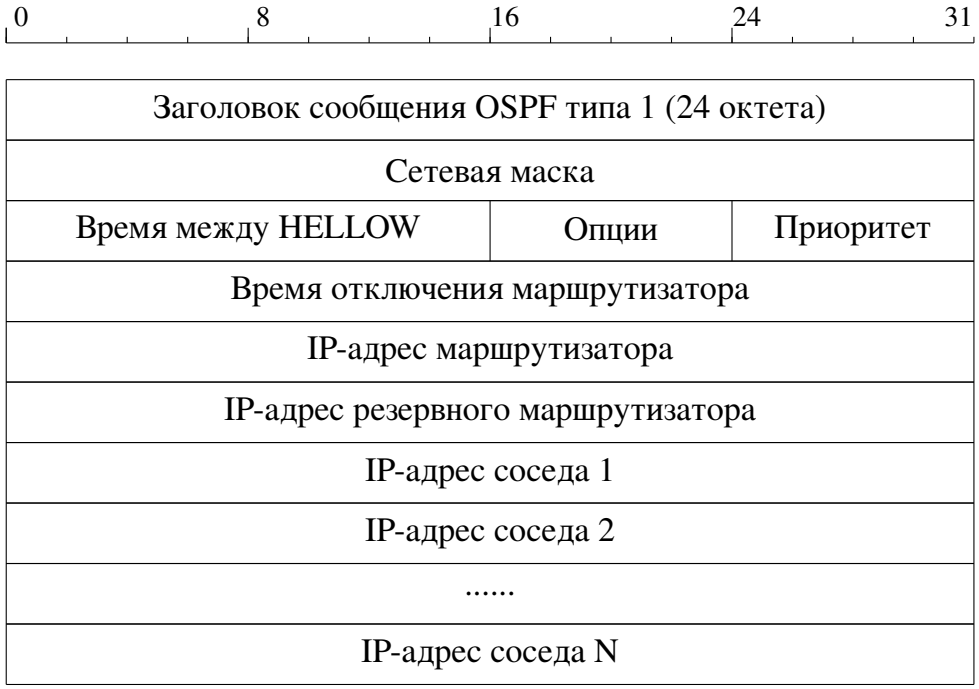


Рис. 5.35. Формат сообщения Hellow протокола OSPF

Поле *Сетевая маска* соответствует маске подсети данного интерфейса.

Поле *Время между сообщениями HELLOW* содержит значение времени в секундах между сообщениями Hellow.

Поле *Приоритет* определяет уровень приоритета маршрутизатора.

Поле *Время отключения маршрутизатора* определяет временной интервал в секундах, по истечении которого не отвечающий маршрутизатор считается вышедшим из строя.

Поля *IP-адрес маршрутизатора* и *IP-адрес резервного маршрутизатора* указывают, куда надо послать сообщение.

Поля *IP-адрес соседа* i образуют список адресов соседних маршрутизаторов, откуда за последнее время были получены сообщения Hellow.

Поле *Опции* (8 бит) информирует о состояние канала и описывает базу данных. Формат поля *Опции* протокола OSPF с типом сообщения Hellow приведён на рис. 5.36.

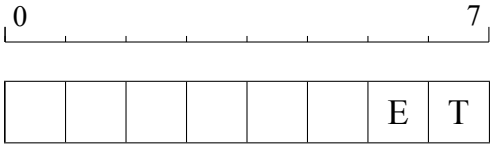


Рис. 5.36. Формат поля *Опции* протокола OSPF с типом сообщения Hellow

Бит *E* характеризует возможность внешней маршрутизации и имеет значение только в сообщениях типа Hellow, в остальных сообщениях данный бит должен быть обнулён (т.е. маршрутизатор не будет посылать или принимать маршрутную информацию от внешних автономных систем).

Бит *T* определяет сервисные возможности маршрутизатора (Type of Service, ToS). Если $T = 0$, то маршрутизатор поддерживает лишь один вид услуг.

Формат сообщения OSPF о маршрутах приведён на рис. 5.37.

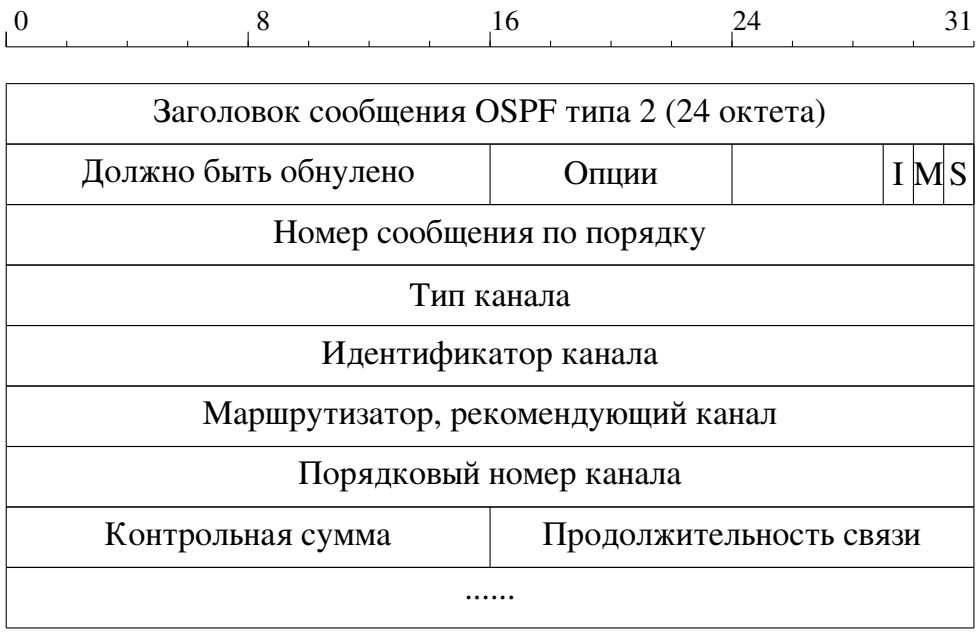


Рис. 5.37. Формат OSPF-сообщения о маршрутах

Поля, начиная с поля *Тип канала*, повторяются для каждого описания канала.

Содержимое базы может пересылаться по частям. В стартовом сообщении бит *I* устанавливается в 1, в сообщении-продолжении бит *M* устанавливается в 1. Бит *S* определяет, послано сообщение сервером ($S = 1$) или клиентом ($S = 0$).

Поле *Номер сообщения по порядку* служит для контроля пропущенных блоков.

Поле *Тип канала* характеризует объявление о маршруте и может принимать следующие значения:

1 — описание каналов маршрутизатора (состояние его интерфейсов);

2 — описание сетевых каналов (перечень маршрутизаторов, непосредственно связанных с сетью);

3 или 4 — сводное описание каналов, в которое входят маршруты между отдельными областями сети (тип 3 приписан маршрутам, ведущим к сетям, а тип 4 — маршрутам, ведущим от сетей);

5 — описание внешних связей автономной системы.

Поле *Идентификатор канала* определяет характер канала (идентификатором может быть IP-адрес маршрутизатора или сети).

Поле *Маршрутизатор, рекомендующий канал* определяет адрес этого маршрутизатора.

Поле *Порядковый номер канала* позволяет маршрутизатору контролировать порядок прихода сообщений и их потерю.

Поле *Продолжительность связи* определяет время в секундах с момента установления связи.

Формат OSPF-запроса маршрутной информации приведён на рис. 5.38.

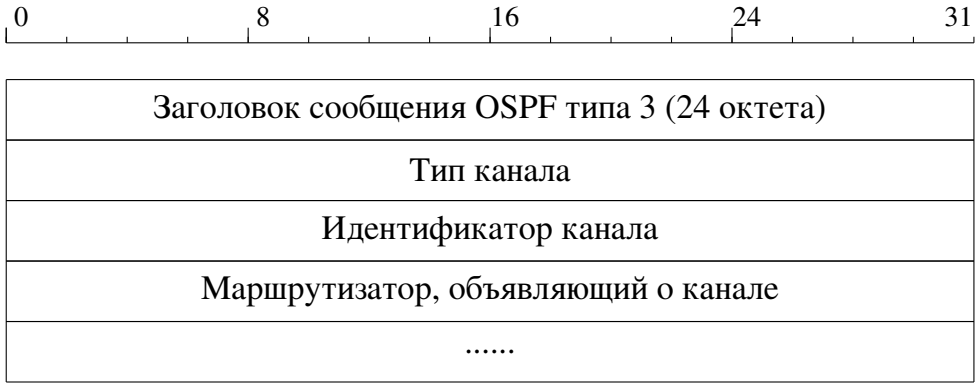


Рис. 5.38. Формат OSPF-запроса маршрутной информации

Формат сообщения о получении OSPF-пакета приведён на рис. 5.39.

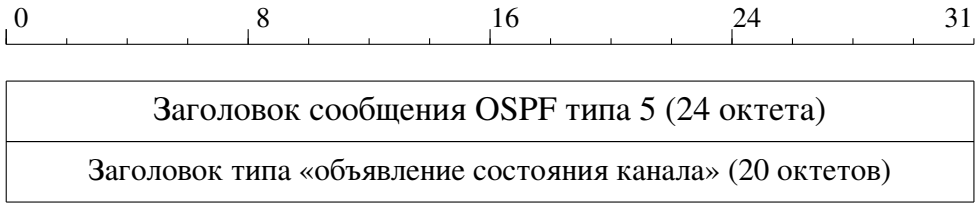


Рис. 5.39. Формат сообщения о получении OSPF-пакета

Формат OSPF-сообщения об изменении маршрутов приведён на рис. 5.40.

Сообщения об изменении маршрута могут быть вызваны следующими причинами:

1) продолжительность связи достигла предельного значения;

2) изменилось состояние интерфейса;
3) произошли изменения в маршрутизаторе сети;
4) произошли изменения в одном из соседних маршрутизаторов;
5) изменилось состояние одного из внутренних маршрутов;
6) изменение состояния межзонного маршрута;
7) появление нового маршрутизатора, подключённого к сети;
8) изменение виртуального маршрута одним из маршрутизаторов сети;
9) изменение одного из внешних маршрутов;
10) маршрутизатор перестал быть пограничным для данной автономной системы.

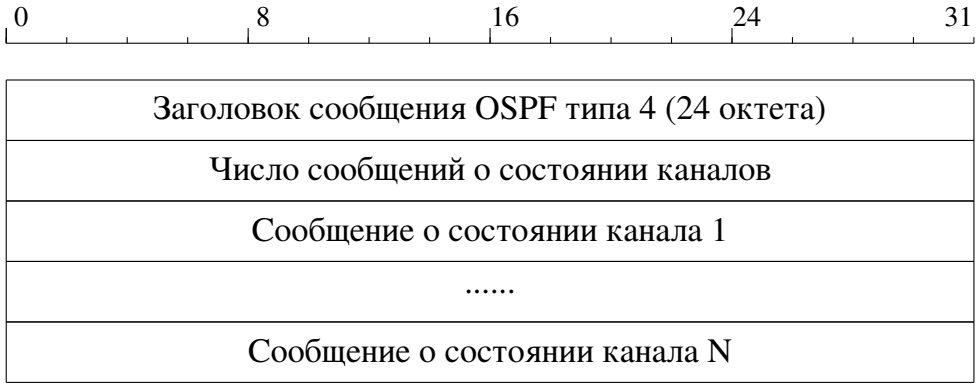


Рис. 5.40. Формат OSPF-сообщения об изменении маршрутов

5.4.2.4.3. Достоинства протокола OSPF. В отличие от универсальных протоколов (например, RIP), протокол OSPF предназначен для построения маршрутов только в сетях TCP/IP.

Основными достоинствами протокола OSPF являются:

— отсутствие ограничений на размер сети;
— поддержка внеклассовых сетей;
— передача сообщений протокола с использованием multicast-адресов, причём отдельные адреса используются для передачи и приёма информации о маршрутах в информационной системе;
— высокая скорость установления маршрутов при изменении состояния системы;
— встроенная процедура установления подлинности источника маршрутной информации;
— возможность использования нескольких параллельных путей к одному пункту назначения (Load Balancing);
— композитная метрика;
— иерархическая маршрутизация.

5.4.2.5. Протокол BGP

Протокол пограничного шлюза (Border Gateway Protocol,BGP) является протоколом маршрутизации между автономными системами. Данный протокол работает поверх протокола транспортного уровня. Это позволяет не нагружать сервисы обработки протокола BGP механизмами фрагментации или обеспечения достоверности доставки пакетов. Схемы аутентификации протоколов транспортного уровня также могут быть использованы BGP в дополнение к собственной системе аутентификации. Кроме того, хотя BGP разработан как протокол маршрутизации между автономными системами, он может использоваться для маршрутизации и внутри автономных систем.

Основным предназначением BGP является обеспечение обмена информацией с другими BGP-системами о досягаемости определённых сетей или хостов. Эта информация должна содержать набор маршрутов к данной сети, т.е. должны быть указаны все промежуточные автономные системы. Такой информации вполне достаточно для того, чтобы построить граф соединений между автономными системами и проконтролировать возможные маршрутные петли. На основании этих данных BGP выбирает оптимальный маршрут и передаёт эту информацию своим соседям.

5.4.2.5.1. Отличия протокола BGP от других протоколов маршрутизации.

Протокол BGP нельзя отнести ни к классу дистанционно-векторных, ни к классу протоколов маршрутизации на основе состояния канала. Ниже приведены характерные отличия протокола BGP от других протоколов маршрутизации.

— *Коммуникация между автономными системами.*
Поскольку протокол BGP относится к протоколам внешнего шлюза, его основное назначение — обеспечить обмен информацией между двумя автономными системами.

— *Координирование работы нескольких спикеров BGP.*
Если в состав автономной системы входит несколько маршрутизаторов, каждый из которых обменивается информацией с равным ему по рангу маршрутизатором внешней автономной системы (их называют *спикерами BGP*), протокол BGP может использоваться для координации работы всего набора маршрутизаторов. Это гарантирует, что маршрутизаторы распространяют непротиворечивую информацию.

— *Распространение информации о достижимости.*
Протокол BGP позволяет автономной системе сообщить информацию о расположенных в ней получателях, а также о тех получателях, доступ к которым осуществляется через данную автономную систему. Кроме того, с помощью протокола BGP подобную информацию можно получить от других автономных систем.

— *Принцип ближайшего перехода.*
Подобно дистанционно-векторным протоколам маршрутизации, протокол BGP предоставляет информацию об адресе *ближайшей точки перехода* для каждого получателя.

— *Поддержка различной политики маршрутизации.*
В отличие от многих дистанционно-векторных протоколов, которые сообщают только ту маршрутную информацию, которая находится в локальной таблице маршрутизации, протокол BGP обеспечивает политику маршрутизации в зависимости от выбора администратора. В частности, маршрутиза-

тор, работающий под управлением протокола BGP, можно настроить так, чтобы он различал получателей, доступ к которым осуществляется через компьютеры его автономной системы, и получателей, анонсированных другими автономными системами.

— *Надёжный транспортный протокол.*
Протокол BGP отличается от других протоколов, передающих информацию о маршрутизации, тем, что он предполагает использование надёжного транспортного протокола. Таким образом, для обмена информацией в протоколе BGP используется исключительно транспортный протокол TCP.

— *Информация о маршруте.*
Кроме указания списка возможных получателей и адреса ближайшей точки перехода для каждого из них в сообщении протокола BGP анонсируется также маршрутная информация, которая позволяет узнать, через какие автономные системы проложен маршрут к конкретному получателю.

— *Передача обновлений.*
Чтобы не создавать дополнительную нагрузку на сеть, в каждом сообщении протокола BGP об обновлении не передаётся полная маршрутная информация. Вместо этого обмен полной информацией происходит только один раз, а в следующих сообщениях передаются только изменения.

— *Поддержка бесклассовой адресации.*
Протокол BGP поддерживает CIDR-адреса. Это означает, что программа протокола BGP не полагается на методы идентификации IP-адресов, а вместе с каждым адресом отсылает и его маску.

— *Объединение маршрутов.*
Чтобы не создавать дополнительной нагрузки на сеть, протокол BGP позволяет отправителю накапливать информацию о маршрутах и отсылать в одном пакете данные сразу о нескольких, связанных между собой получателях.

— *Аутентификация.*
Протокол BGP позволяет получателю удостоверить подлинность сообщений (т.е. подтвердить «личность» отправителя).

5.4.2.5.2. Функции протокола BGP и виды сообщений. В процессе взаимодействия по протоколу BGP выполняется три основных действия:

1) получение согласия сторон на взаимодействие по протоколу BGP и аутентификацию (при этом два равноправных маршрутизатора устанавливают соединение по протоколу TCP и обмениваются сообщениями, которые подтверждают, что обе стороны согласны вступить в процесс обмена информацией);
2) каждая сторона отсылает информацию о доступности или недоступности получателей (это означает, что отправитель может сообщить о возможности доступа к одной или нескольким сетям получателя (при этом указывается адрес ближайшей точки перехода для каждой сети) или, напротив, может заявить, что одна или несколько сетей, о которых сообщалось ранее, более недоступны);
3) осуществление постоянного контроля над правильностью функционирования взаимодействующих пар маршрутизаторов и сетевых соединений.

В протоколе BGP определено четыре основных типа сообщений: OPEN (инициализирует процесс), UPDATE (аннулирует маршрутную информацию), NOTI-

FICATION (отвечает на неверное сообщение, KEEPALIVE (выполняет активную проверку возможности соединения между BGP-парами).

В начале каждого сообщения протокола BGP расположен заголовок фиксированного формата, с помощью которого определяется тип сообщения (рис. 5.41).

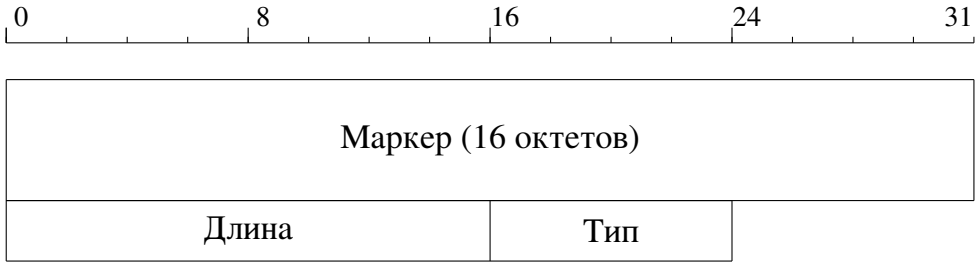


Рис. 5.41. Формат сообщения BGP

В поле *Маркер (Marker)* (16 октетов) заносится значение, которое обе стороны «договорились» использовать в качестве метки начала сообщения.

В поле *Длина (Length)* (2 октета) указывается общая длина сообщения в октетах. Минимальный размер сообщения составляет 19 октетов (для типа сообщения, в котором после заголовка нет данных). Максимально допустимая длина сообщения составляет 4096 октетов.

Наличие поля маркера является нехарактерным для сетевых протоколов. В исходном сообщении маркер состоит из всех единиц. Если взаимодействующие между собой маршрутизаторы «договорятся» об использовании механизма аутентификации, в поле маркера может содержаться информация об аутентификации. В любом случае обе стороны должны согласовать, какое значение будет внесено в это поле, чтобы его можно было в дальнейшем использовать для выполнения синхронизации.

Обмен всеми типами сообщений в протоколе BGP происходит через протокол TCP, в котором невозможно определить, где заканчивается одно сообщение и начинается другое. В такой среде ошибка, произошедшая на стороне одного из участников соединения, может привести к потере пакета, а получатель никогда не узнает об ошибке. Таким образом, чтобы обеспечить синхронные действия отправителя и получателя, BGP помещает в начало каждого сообщения некоторую известную обеим сторонам последовательность октетов, и перед дальнейшей обработкой сообщения требует от получателя подтвердить, что данное значение не повреждено.

BGP-сообщения OPEN (рис. 5.42) является запросом BGP-соединения и передаётся для организации сеанса связи между равноправными BGP-маршрутизаторами. Принявший это сообщение маршрутизатор подтверждает установление соединения, передавая сообщение KEEPALIVE (сообщение KEEPALIVE содержит только заголовок и обеспечивает сброс таймера удержания соединения).

Поле *Версия* (длина 8 бит) указывает на версию протокола BGP.

Поле *Моя автономная система* (длина 16 бит) содержит идентификатор автономной системы отправившего сообщение маршрутизатора.

Поле *Время сохранения (Hold Time)* (длина 16 бит) указывает на максимальное время (в секундах) между приходами сообщений KEEPALIVE, используемых для монитронига активности соединения.

Поле *BGP-идентификатор* (длина 32 бита) содержит идентификатор маршрутизатора (один из адресов интерфейсов).

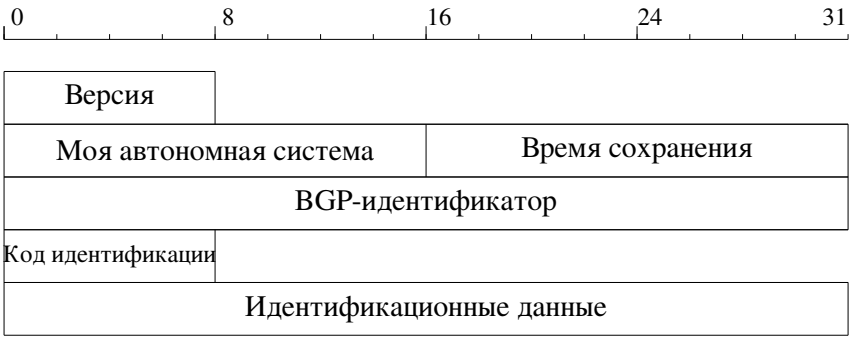


Рис. 5.42. Формат BGP-сообщения OPEN

Поле *Код идентификации* (длина 8 бит) содержит длину поля *Идентификационные данные*, содержащее различные опции.

Сообщение UPDATE рассылается маршрутизатором BGP с целью внесения изменений в таблицы маршрутизации. Формат BGP-сообщения UPDATE приведён на рис. 5.43. Сообщение UPDATE состоит из трех частей переменной дли-

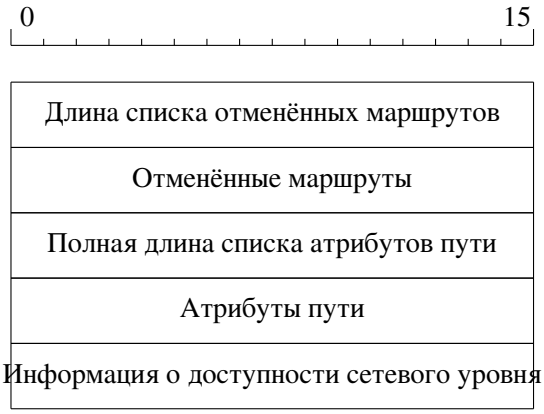


Рис. 5.43. Формат BGP-сообщения Update

ны: списка отменённых (недействительных) маршрутов, списка атрибутов пути и списка сетей, к которым эти атрибуты относятся. Две последние части представляют собой собственно информацию о маршруте в указанные сети.

Список адресов сетей и список недействительных маршрутов представляют собой списки элементов, состоящих из длины префикса и собственно сетевого адреса.

Сообщениями-уведомлениями (NOTIFICATION) BGP-маршрутизаторы обмениваются при возникновении ошибок. Такие сообщения содержат в себе код ошибки (например, ошибка заголовка, ошибка в сообщении OPEN, ошибка в сообщении UPDATE и т.д.).

5.5. Коммутация пакетов по меткам (MPLS)

Технология коммутации пакетов по меткам в многопротокольных сетях (Multiprotocol Label Switching, MPLS) представляет собой механизм передачи данных,

который эмулирует различные свойства сетей с коммутацией каналов поверх сетей с коммутацией пакетов (RFC 3031 [27]).

В традиционной IP-сети при передаче пакетов маршрутизаторы на основе данных заголовков (адрес назначения) принимают решение о выборе дальнейшего маршрута.

В сетях на базе протокола MPLS заголовки передаваемых пакетов не анализируются при прохождении через маршрутизаторы, а переадресация осуществляется исключительно на основе меток.

5.5.1. Архитектура MPLS

В основе архитектуры MPLS, как следует из названия, лежит процесс коммутации пакетов по меткам. *Метка (Label)* представляет собой короткий идентификатор фиксированной длины, который определяет принадлежность пакета к некоторому классу на каждом из участков коммутируемого маршрута.

Сеть MPLS делится на две области — *ядро* и *граничную область* (рис. 5.44).

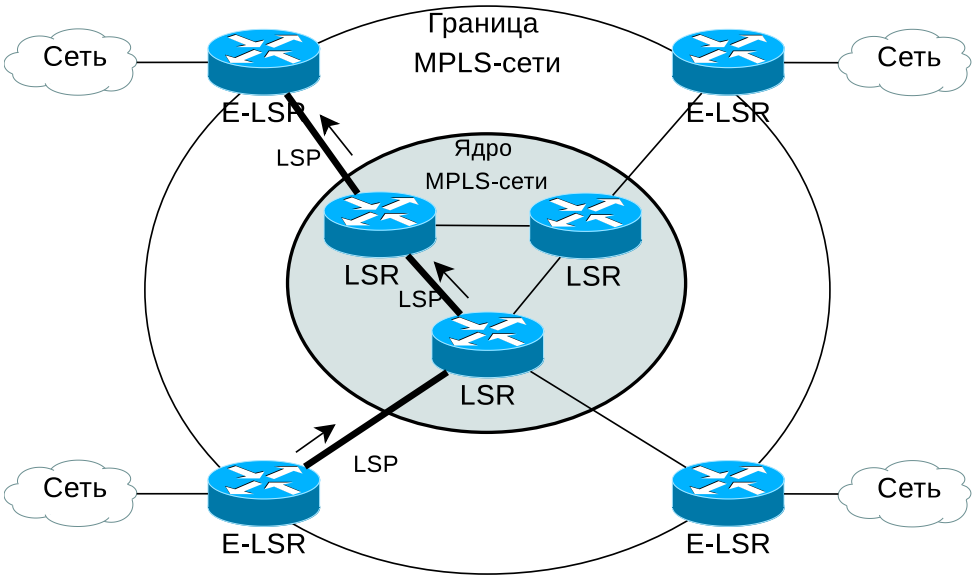


Рис. 5.44. Архитектура сети MPLS

Ядро образуют устройства *Label-Switch Routers (LSR)* — маршрутизаторы, поддерживающие как обычную IP-маршрутизацию, так и коммутацию по меткам. Маршрутизаторы ядра отвечают только за коммутацию. Границу сети MPLS образуют *граничные маршрутизаторы (Edge LSR, E-LSR)*, осуществляющие классификацию поступающих в MPLS-сеть пакетов, их фильтрацию, управление трафиком и т.п. Первая метка, устанавливаемая на граничном маршрутизаторе, определяет *маршрут следования (Label Switch Path, LSP)* пакета через MPLS-домен.

Множество подсетей, поставленное в соответствие конкретному LSP, образуют *класс эквивалентности (Forwarding Equivalence Classes, FEC)*. Каждый из классов FEC обрабатывается отдельно — строится свой путь LSP, выделяется своя ширина полосы пропускания канала и т.п.

LSR выполняет две функции — *маршрутизацию* и *коммутацию по меткам*.

Процесс маршрутизации функционирует на базе внутреннего протокола маршрутизации (например, OSPF). LSR получает маршрутную информацию от соседних маршрутизаторов и формирует таблицу маршрутизации, которая используется для маршрутизации IP-пакетов.

Процесс коммутации функционирует на базе *протокола обмена метками (Label Distribution Protocol, LDP)*, ставящего в соответствие конкретному значению метки определённый маршрут LSP.

5.5.2. Формат MPLS-метки

На рис. 5.45 представлен формат MPLS-метки (RFC 3032 [28]).

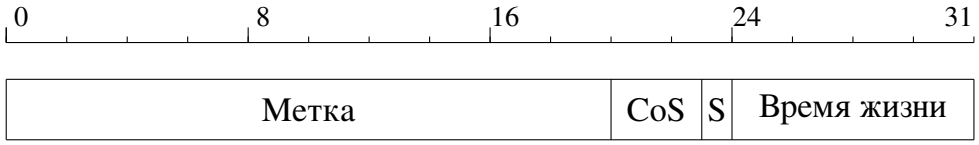


Рис. 5.45. Формат MPLS-метки

Поле *Метка (Label)* (длина 20 бит) содержит код метки, по которой осуществляется коммутация.

Зарезервированные значения меток:

— 0 (IPv4 Explicit NULL Label) — указывает, что стек меток должен быть очищен, а переадресация пакета должна основываться на заголовке IPv4;
— 1 (Router Alert Label) — указывает на то, что переадресация пакета определяется меткой;
— 2 (IPv6 Explicit NULL Label) — указывает, что стек меток должен быть очищен, а переадресация пакета должна основываться на заголовке IPv6;
— 3 (Implicit NULL Label) — значение, присваиваемое маршрутизатором.

Поле *(Class of Service, CoS)* (длина 3 бита) характеризует класс обслуживания пакета.

Поле *S* может принимать значение 0 или 1, указывая, является ли метка последней в стеке меток, присвоенных одному пакету[1].

Поле *Время жизни (Time-to-Live, TTL)* (длина 8 бит) указывает в общем случае число возможных промежуточных узлов.

MPLS-метка передаётся в составе любого пакета, причём способ её присоединения к пакету зависит от используемой технологии канального уровня. MPLS-метка добавляется между заголовком кадра (второй уровень ISO/OSI) и заголовком пакета (третий уровень модели ISO/OSI) (рис. 5.46).

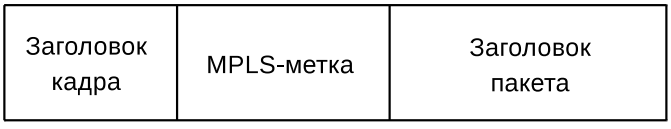


Рис. 5.46. Расположение MPLS-метки

[1] В рамках архитектуры MPLS вместе с пакетом разрешено передавать не одну метку, а целый стек.

5.5.3. Label Distribution Protocol

Протокол распространения меток (Label Distribution Protocol, LDP) предназначен для построения целостных маршрутов LSP (RFC 3036 [29]). LDP представляет собой набор процедур и сообщений, с помощью которых LSR формирует сетевой маршрут LSP путём установления соответствия между маршрутной информацией и каналами передачи данных.

В функции LDP входит: определение соседнего маршрутизатора, управление сессией, рассылка меток, уведомление об ошибках.

Обмены сообщениями LDP осуществляются путём посылки протокольных данных LDP (PDU) через LDP-секцию TCP-соединений. При этом каждый LDP PDU может содержать более одного LDP-сообщения. Каждый LDP PDU представляет собой LDP-заголовок (рис. 5.47), за которым следует одно или более LDP-сообщений.

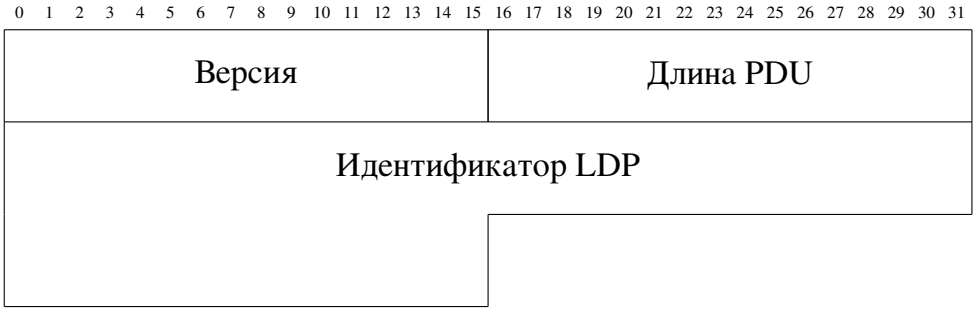


Рис. 5.47. Заголовок LDP

Поле *Версия (Version)* (длина 2 байта) содержит код номера версии протокола.

Поле *Длина PDU (PDU Length)* (длина 2 байта) указывает общую длину PDU в октетах, исключая поля версии и длины PDU.

Поле *Идентификатор LDP (LDP Identifier)* (длина 6 байт) однозначно идентифицирует пространство меток LSR-отправителя. При этом первые четыре октета идентифицируют LSR и должны быть уникальными, а последние два октета идентифицируют пространство меток заданного LSR.

Все сообщения LDP имеют определённый формат (рис. 5.48).

Рис. 5.48. Формат LDP-сообщений

Поле U представляет собой бит неизвестного сообщения; при $U = 1$ сообщение игнорируется.

Поле *Тип сообщения (Message Type)* идентифицирует тип сообщения.

Поле *Длина сообщения (Message Length)* указывает суммарную длину в октетах полей идентификатора сообщения, обязательных параметров и опций.

Поле *Идентификатор сообщения (Message ID)* идентифицирует сообщение.

Поле *Обязательные параметры* представляет собой набор необходимых параметров.

Поле *Опции* представляет собой набор необязательных параметров.

В LDP определены следующие типы сообщений:

— *Hello* — определение соседнего маршрутизатора;
— *инициализация (Init)* — процедура установления сессии;
— *KeepAlive* — используется для поддержания активного статуса LDP-сессии;
— *адрес (Address Message)* — анонсирование адреса интерфейса маршрутизатора;
— *отзыв адреса (Address Withdraw)* — отзыв ранее анонсированного адреса интерфейса;
— *присвоение метки (Label Mapping)* — сообщение о присвоении метки;
— *запрос метки (Label Request)* — запрос метки у соседнего маршрутизатора с целью установления соответствия значения метки и FEC;
— *запрос ликвидации метки (Label Release)* — подтверждение получения метки в сообщении Label Mapping;
— *отзыв метки (Label Abort Request)* — сигнал соседнему маршрутизатору о невозможности продолжения использования ассоциации FEC–метка;
— *освобождение метки (Label Withdraw)* — сообщение о ненужности ранее полученной метки.

Установление LDP сессии происходит по следующему сценарию:

— при помощи обмена сообщениями *Hello* соседние маршрутизаторы определяют транспортные адреса друг друга;
— один из маршрутизаторов становится активным;
— активный маршрутизатор устанавливает TCP/IP сессию на порт 646 и посылает сообщение *Init*, включающее в себя параметры LDP-сессии;
— пассивный маршрутизатор проверяет полученные параметры LDP-сессии на совместимость с локальными настройками LDP и посылает ответное сообщение *Init* со своими параметрами LDP-сессии;
— активный маршрутизатор также проверяет полученные параметры LDP-сессии на совместимость с локальными настройками LDP, после чего сессия считается установленной.

Если на каком-то этапе возникают ошибки, то сессия считается неустановленной, а маршрутизатор, обнаруживший ошибку, посылает сообщение *Shutdown* или *Reject* своему соседу.

LDP-сессия будет установлена, если совпадают версии протокола LDP и совпадают режимы распространения информации о метках.

5.5.4. Сервисы на базе MPLS

На базе MPLS возможна организация следующих сервисов:

— MPLS/VPN — создание распределённых виртуальных частных сетей (Virtual Private Network, VPN) на крупных сетях без организации туннелей и шифрования;

— MPLS/TrafficEnginering — гибкое управление потоками трафика внутри MPLS-домена и более полное использование канальной инфраструктуры сети;
— AnyTransportOverMPLS — прозрачная передача через MPLS-домен кадров ATM, Frame Relay, Ethernet и т.п.

5.5.5. Особенности MPLS

Главной особенностью MPLS является отделение процесса коммутации пакета от анализа IP-адресов в его заголовке. Вся информация о маршруте содержится в метке, и пакету не требуется нести адреса промежуточных узлов, что улучшает управление распределением нагрузки в сети.

В сетях MPLS есть возможность организации при помощи протокола RSVP явной коммутации пакетов через так называемые туннели, что повышает эффективность загрузки каналов в MPLS-сети с альтернативными путями, поскольку трафик с определённой меткой идёт по конкретному пути с заданными параметрами качества обслуживания. Такое решение снимает необходимость иметь маршрутную информацию на всех маршрутизаторах в сети оператора.

Ещё одной важной особенностью сетей MPLS является возможность разделения IP-трафика и создания VPN-соединений между различными узлами, а также независимость адресных пространств операторской и клиентских сетей. Такое решение даёт возможность масштабирования сети, интеграции сети с другими сервисами IP.

Глава 6. Транспортный уровень

На транспортном уровне организована служба надёжной доставки данных для верхних уровней, использующая управление потоком и коррекцию ошибок в сквозном потоке. Некоторые реализации транспортного уровня дополнительно осуществляют сегментацию данных при их отправке и воссоздании на приёмной стороне. Кроме того, транспортный уровень предоставляет приложению одно или несколько виртуальных соединений, связывающих оконечные точки.

Сегментация данных позволяет разделить большой блок данных, переданных приложением, на более мелкие фрагменты, которые способен передать сетевой уровень. На сетевом уровне выполняется инкапсуляция заголовков пакетов транспортного протокола и прикладных данных, а сформированный пакет передаётся на канальный уровень.

Управление потоком на транспортном уровне обычно сопровождается ограничением числа пакетов, которые могут быть посланы без подтверждения их приёма.

На транспортном уровне семейства протоколов TCP/IP применяются два основных протокола — ориентированный на соединение протокол TCP и не требующий соединения протокол UDP.

Важной концепцией служб транспортного уровня семейства протоколов TCP/IP является концепция *портов*, представляющих собой 16-битный номер и идентифицирующих службу прикладного уровня стека протоколов TCP/IP.

6.1. Протокол UDP

Относить *протокол пользовательских датаграмм (User Datagram Protocol, UDP)* (см. RFC 768 [30]) к транспортному уровню не вполне корректно. *UDP* ненадёжен в том смысле, что доставка пакетов не гарантируется. Протокол *UDP* — это протокол *без установления соединения (ConnectionLess)*. Он не устанавливает виртуального соединения, не осуществляет никаких повторных передач, не выполняет переупорядочивания пакетов, не управляет потоком данных. Все эти функции возложены на протоколы более высокого уровня (или приложения).

Формат заголовка пакета UDP показан на рис. 6.1. Поле данных на рисунке не показано. Заметим лишь, что оно выравнивается по 32-битной границе нулевыми байт-заполнителями.

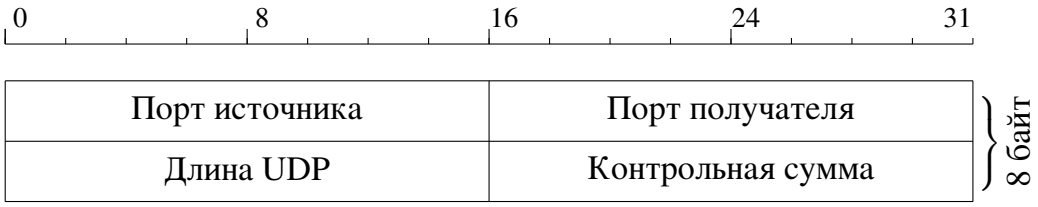


Рис. 6.1. Формат заголовка пакета UDP

Поля *Порт источника (Source Port)* (длина 16 бит) и *Порт получателя (Destination Port)* (длина 16 бит) идентифицируют передающий и получающий процессы соответственно.

Поле *Длина UDP (Length)* (длина 16 бит) содержит длину пакета UDP в байтах.

Поле *Контрольная сумма UDP (Checksum)* (длина 16 бит) содержит контрольную сумму пакета UDP, вычисляемую по всему пакету UDP с добавленным псевдозаголовком (рис. 6.2).

Рис. 6.2. Структура пакета UDP при вычислении контрольной суммы

Во время вычисления контрольной суммы это поле выставляется в нуль, а поле данных выравнивается по 32-байтной границе нулевыми байтами. Если контрольная сумма в полученном пакете равняется нулю, то считается, что передающий уровень UDP её не вычисляет, и данные не защищены.

Псевдозаголовок формируется исключительно для работы с контрольной суммой и имеет следующую структуру (рис. 6.3).

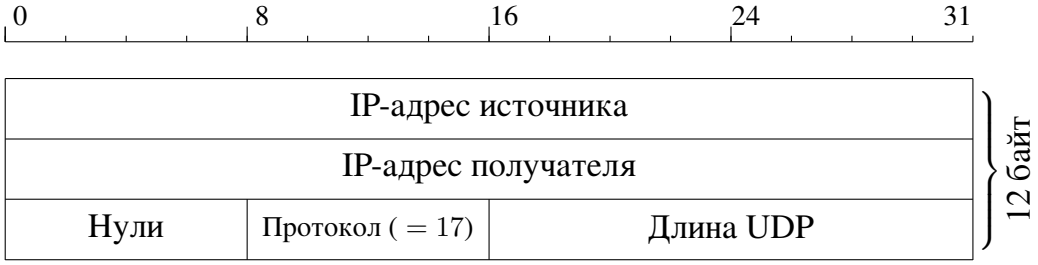


Рис. 6.3. Структура псевдозаголовка пакета UDP

Вначале идут поля *IP-адрес источника* (длина 32 бит) и *IP-адрес получателя* (длина 32 бит).

Далее идёт зарезервированное поле (длина 8 бит), заполненное нулями.

Поле *Протокол* (длина 8 бит) идентифицирует протокол из заголовка пакета IP. Для UDP это значение равно 17 (см. табл. 5.1).

Далее идёт поле *Длина UDP* (длина 16 бит).

Защита заголовка IP несколько избыточна и делает протокол UDP (впрочем, как и TCP) неотделимым от протокола IP, хотя это и позволяет провести двойную проверку датаграмм IP, поступивших для заданного получателя.

Протоколом UDP пользуются приложения, которым нужно передавать датаграммы последовательно. Например, это такие протоколы, как *протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP)*, *служба именования доменов (Domain Name Service, DNS)*, *простой протокол управления сетью (Simple Network Management Protocol, SNMP)* и др. Пользуясь UDP, приложение несёт ответственность за коррекцию ошибок.

6.2. Протокол TCP

Протокол управления передачей (Transmission Control Protocol, TCP) является, в отличие от UDP, «настоящим» протоколом транспортного уровня, который имеет средства управления потоком и коррекции ошибок. Он ориентирован на установление соединения, поэтому клиент обязан установить соединение с сервером до начала передачи данных TCP в любом из направлений (RFC 793 [31])[1].

6.2.1. Формат пакета TCP

На рис. 6.4 показана структура заголовка сегмента TCP.

Рис. 6.4. Формат заголовка пакета TCP

Поля *Порт источника (Source Port)* (длина 16 бит) и *Порт получателя (Destination Port)* (длина 16 бит) аналогичны таким же полям в заголовке пакета UDP (см. раздел 6.1) и идентифицируют процесс или приложение, использующее протокол TCP.

Поля *Порядковый номер (Sequence Number)* (длина 32 бита) и *Номер подтверждения (Acknowledgement Number)* (длина 32 бита) нумеруют каждый отправленный или полученный байт данных. Эти поля реализуются как целые числа без знака, которые сбрасываются, когда достигают максимального значения. Каждая сторона ведёт собственную порядковую нумерацию.

Поле *Длина заголовка (Offset)* (длина 4 бита) содержит размер TCP-заголовка в 32-битных словах. Эта информация необходима, так как поле *Параметры (Option)* может быть переменной длины. Можно сказать, что это поле задаёт смещение от начала сегмента до начала данных в 32-битных словах.

Следом идёт неиспользуемое поле (Resrvd) длиной 6 бит.

Затем идёт поле *Флаги* (длина 6 бит), содержащее шесть 1-битовых флагов (рис. 6.5).

Флаг *Указатель срочности (Urgent Pointer, URG)* устанавливается в 1 в случае использования поля *Указатель на срочные данные*.

[1]Также в следующих RFC: 1323, 1644, 2018, 2581, 2582, 2861, 2873, 2883, 2923, 2988, 3293, 3448, 3465, 3481.

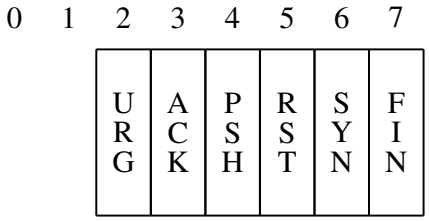


Рис. 6.5. Поле *Флаги* заголовка пакета TCP

Флаг *Подтверждение (Acknowledgment, ACK)* устанавливается в 1 в случае, если поле *Номер подтверждения (Acknowledgement Number)* содержит данные. В противном случае это поле игнорируется.

Флаг *Выталкивание (Push, PSH)* означает, что принимающий стек TCP должен немедленно информировать приложение о поступивших данных, а не ждать, пока буфер заполниться. Большинство современных реализаций TCP просто игнорируют флаг *PSH* во время приёма пакетов. Этот флаг оставлен по историческим причинам.

Флаг *Сброс (Reset, RST)* используется для отмены соединения из-за ошибки приложения, отказа от неверного сегмента, попытки создать соединение при отсутствии затребованного сервиса.

Флаг *Синхронизация (Synchronize, SYN)* устанавливается при инициировании соединения и синхронизации порядкового номера.

Флаг *Завершение (Finished, FIN)* используется для разрыва соединения. Он указывает, что отправитель закончил передачу данных.

Управление потоком в протоколе TCP осуществляется при помощи скользящего окна переменного размера. Поле *Размер окна (Window)* (длина 16 бит) содержит количество байт, которое может быть послано после байта, получение которого уже подтверждено. Если значение этого поля равно нулю, это означает, что все байты, вплоть до байта с номером *Номер подтверждения* – 1, получены, но получатель отказывается принимать дальнейшие данные. Разрешение на дальнейшую передачу может быть выдано отправкой сегмента с таким же значением поля *Номер подтверждения* и ненулевым значением поля *Размер окна*.

Поле *Контрольная сумма TCP (Checksum)* (длина 16 бит) содержит контрольную сумму пакета TCP, вычисляемую по всему пакету TCP с добавленным псевдозаголовком (рис. 6.6). Во время вычисления контрольной суммы это поле выставляется в нуль, а поле данных выравнивается по 32-байтной границе нулевыми байтами.

Рис. 6.6. Структура пакета TCP при вычислении контрольной суммы

Псевдозаголовок формируется исключительно для работы с контрольной сум-

мой и имеет следующую структуру (рис. 6.7).

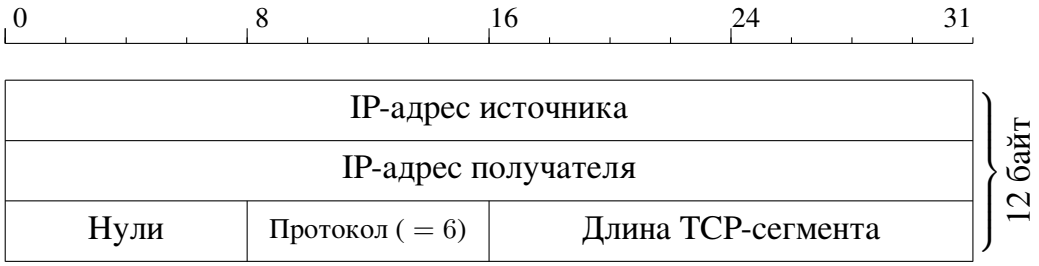


Рис. 6.7. Структура псевдозаголовка пакета TCP

Вначале идут поля *IP-адрес источника* (длина 32 бит) и *IP-адрес получателя* (длина 32 бит).

Далее идёт зарезервированное поле (длина 8 бит), заполненное нулями.

Поле *Протокол* (длина 8 бит) идентифицирует протокол из заголовка пакета IP. Для TCP это значение равно 6 (см. табл. 5.1).

Далее идёт поле *Длина TCP* (длина 16 бит).

Поле *Указатель на срочные данные* (длина 16 бит) содержит смещение в байтах от текущего порядкового номера байта до места расположения срочных данных. Содержимым срочных данных занимаются вышестоящие уровни.

Поле *Параметры (Option)* (длина переменная, кратная 32 битам) содержит дополнительные поля, расширяющие возможности стандартного заголовка. Это поле зарезервировано для будущего применения и в заголовке может отсутствовать. В настоящее время определены опции:

— конец списка опций;
— никаких операций (используется для заполнения поля опции до числа октетов, кратного 4);
— максимальный размер сегмента (Maximum Segment Size, MSS), задающий верхний размер поля данных.

Данные в TCP-сегменте могут и отсутствовать, характер и формат передаваемой информации задаются исключительно прикладной программой, теоретически максимальный размер этого поля составляет в отсутствие опций 65495 байт.

6.2.2. Установление сессии TCP

Поля *Порядковый номер (Sequence Number)* и *Номер подтверждения (Acknowledgment Number)* играют роль счётчика пакетов. При установлении сессии используется поле флагов.

Установление связи клиент-сервер осуществляется в три этапа (трёхступенчатый handshake) (рис. 6.8) .

Пусть хост A создаёт соединение с хостом B.

1) *Режим активного доступа (Active Open)*. Клиент посылает сообщение *SYN, ISSa*, т.е. в передаваемом сообщении установлен бит SYN (Synchronize Sequence Number), а в поле *Порядковый номер (Sequence Number)* — начальное 32-битное значение ISSa (Initial Sequence Number).
2) *Режим пассивного доступа (Passive Open)*. Сервер откликается, посылая сообщение *SYN, ACK, ISSb, ACK(ISSa+1)*, т.е. установлены биты SYN и

ACK; в поле *Порядковый номер (Sequence Number)* хостом B устанавливается начальное значение счётчика — ISSb; поле *Номер подтверждения (Acknowledgment Number)* содержит значение ISSa, полученное в первом пакете от хоста A и увеличенное на единицу.

3) *Завершение рукопожатия*. Клиент отправляет подтверждение получения SYN-сегмента от сервера с идентификатором, равным ISN (сервера)+1: *ACK, ISSa+1, ACK(ISSb+1)*. В этом пакете установлен бит ACK, поле *Порядковый номер (Sequence Number)* содержит ISSa + 1, поле *Номер подтверждения (Acknowledgment Number)* содержит значение ISSb + 1. Посылкой этого пакета заканчивается трёхступенчатый handshake, и TCP-соединение считается установленным.
4) Теперь клиент может посылать пакеты с данными на сервер по только что созданному виртуальному TCP-каналу: *ACK, ISSa+1, ACK(ISSb+1); DATA*.

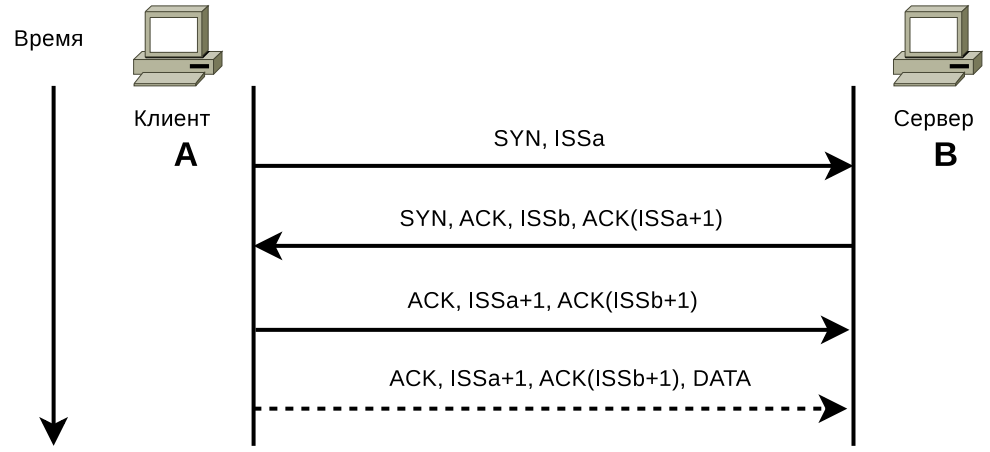


Рис. 6.8. Трёхступенчатый handshake

Из рассмотренной выше схемы создания TCP-соединения видно, что единственными идентификаторами TCP-абонентов и TCP-соединения являются два 32-битных параметра *Порядковый номер (Sequence Number)* и *Номер подтверждения (Acknowledgment Number)*.

6.2.3. Управление потоком

Для ускорения и оптимизации процесса передачи больших объёмов данных протокол TCP определяет метод управления потоком, называемый *методом скользящего окна*, который позволяет отправителю посылать очередной сегмент, не дожидаясь подтверждения о получении в пункте назначения предшествующего сегмента.

Протокол TCP формирует подтверждения не для каждого конкретного успешно полученного пакета, а для всех данных от начала посылки до некоторого порядкового номера *ACK SN (Acknowledge Sequence Number)*. В качестве подтверждения успешного приёма, например, первых n байт, высылается ACK SN = $n+1$: это означает, что все данные в байтовом потоке под номерами от ISN+1=1 до n успешно получены.

Вместе с посылкой отправителю порядкового номера ACK SN получатель объявляет также размер окна. Это значит, что отправитель может посылать дан-

ные с порядковыми номерами от текущего ACK SN до (ACK SN + размер окна - 1), не дожидаясь подтверждения со стороны получателя. Если не будет получено новое подтверждение (новый ACK SN), отправитель будет посылать данные, пока он остаётся в пределах объявленного окна. После этого посылка данных будет прекращена до получения очередного подтверждения и нового размера окна.

Размер окна выбирается таким образом, чтобы подтверждения успевали приходить вовремя и остановки передачи не происходило. Размер окна может динамически изменяться получателем.

Для временной остановки посылки данных достаточно объявить нулевое окно. Но даже в этом случае через определённые промежутки времени будут отправляться сегменты с одним октетом данных. Это делается для того, чтобы отправитель гарантированно узнал о том, что получатель вновь объявил ненулевое окно, поскольку получатель обязан подтвердить получение пробных сегментов, а в этих подтверждениях он укажет также и текущий размер своего окна. В протоколе TCP скользящее окно используется для регулировки трафика и препятствия переполнению буфера.

Регулирование трафика в TCP подразумевает существование двух независимых процессов: *контроля доставки*, управляемого получателем с помощью параметра *Размер окна (Window)*, и *контроля перегрузки*, управляемого отправителем с помощью *Окна перегрузки (Congestion Window, CWnd)* и *Порога медленного старта (Slow Start Threshold, SSThreth)*.

Первый процесс отслеживает заполнение входного буфера получателя, второй — регистрирует перегрузку канала и связанные с этим потери, а также понижает интенсивность трафика. В исходный момент времени при установлении соединения CWnd делается равным одному MSS (максимальному размеру сегмента), а SSThreth — 65535 байтам. Программа, управляющая пересылкой, никогда не пошлёт больше байт, чем это задано CWnd и объявленным получателем значением *Размера окна (Window)*. Когда получение очередного блока данных подтверждено, значение CWnd увеличивается. Если значение CWnd меньше или равно значению SSThreth, то выполняется процедура *Медленный старт*, в противном случае осуществляется подавление перегрузки. В последнем случае CWnd_i+1 = CWnd_i + MSS/8 +(MSS*MSS)/CWnd. Если возникает состояние перегрузки канала, значение CWnd снова делается равным одному MSS. Окно перегрузки позволяет согласовать полную загрузку виртуального соединения и текущие возможности канала, минимизируя потери пакетов при перегрузке.

Для управления потоком используется *Порог медленного старта (SSThresh)*. При установлении соединения SSThresh=64 Кбайт. В случае возникновения таймаута значение SSThresh становится равным CWnd/2, а само значение CWnd приравнивается MSS. Далее запускается процедура медленного старта, чтобы выяснить возможности канала. При этом экспоненциальный рост CWnd осуществляется вплоть до значения SSThresh. Когда этот уровень CWnd достигнут, дальнейший рост происходит линейно с приращением на каждом шаге, равном MSS.

6.2.4. Проблемы TCP

TCP за годы существования претерпел значительные изменения, касающиеся обеспечения надёжности и производительности в сетях различной ёмкости и качества. Но при этом возможности TCP уже не удовлетворяют современным потребностям.

— TCP не подходит для передачи данных в VoIP-сетях или для асинхронной обработки на базе транзакций.
— TCP требует строго упорядоченной передачи данных, что не подходит для приложений, допускающих как последовательную, так и непоследовательную доставку потоков.
— TCP не структурирует последовательности передаваемых данных. Поэтому требуется добавление разграничителей сообщений.
— TCP не поддерживает множественную адресацию.
— TCP-хосты восприимчивы к атакам «отказ в обслуживании» (Denial of Service, DoS) типа SYN DoS (или FIN DoS)[1].

6.3. Протокол SCTP

Протокол управления потоковой передачей (Stream Control Transmission Protocol, SCTP) (RFC 2960 [32], RFC 3286 [33]) можно рассматривать как дальнейшее логическое развитие протокола TCP. Как и TCP, протокол SCTP предлагает приложениям, взаимодействующим по IP-сети, ориентированную на соединения типа «точка-точка» транспортную службу с надёжной доставкой. Протокол унаследовал часть функциональности TCP, в том числе возможность контроля перегрузки и восстановления утерянных пакетов. Любое приложение, работающее по протоколу TCP, можно перевести на SCTP без потери функциональности.

6.3.1. Формат пакета SCTP

Сообщения SCTP включают общий заголовок, за которым следует один или несколько *подпакетов (Chunk)*, которые могут содержать данные или управляющую информацию (рис. 6.9). В заголовке (рис. 6.10) указываются номера портов отправителя и получателя, что позволяет мультиплексировать различные ассоциации SCTP на одном адресе.

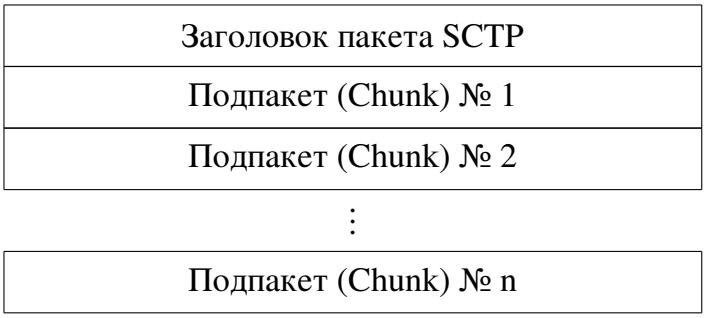


Рис. 6.9. Формат пакета SCTP

[1] Хосту посылается огромное количество пакетов TCP SYN. Хост-получатель резервирует память и отвечает на запрос сообщениями SYN ACK. Когда атакующая система не возвращает сообщения ACK, необходимые для завершения процедуры установки TCP-соединения, ресурсы хоста, подвергнувшегося атаке, остаются неосвобождёнными. Поэтому он оказывается не готов к обслуживанию других запросов.

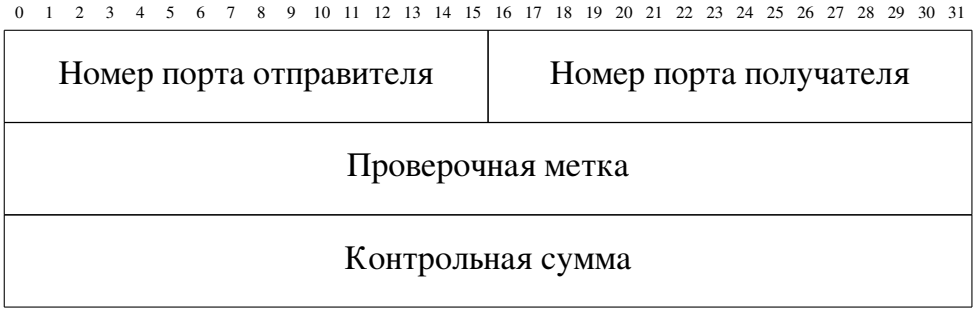


Рис. 6.10. Формат заголовка пакета SCTP

Проверочная метка (Verification Tag) (длина 32 бита) предотвращает возможность включения в ассоциацию SCTP устаревших или фальсифицированных сообщений.

Контрольная сумма (длина 32 бита) рассчитывается на основе полиномиального алгоритма CRC-32c и служит для выявления ошибок.

6.3.1.1. Формат подпакета

Каждый подпакет (фрагмент) содержит поля (рис. 6.11) *Тип подпакета (Chunk ID)*, *Флаги (Chunk Flags)*, *Длина подпакета (Chunk Length)*, *Данные (Chunk Value)*.

Рис. 6.11. Формат подпакета SCTP

Восьмибитное поле типа подпакета способно принимать до 255 значений (в настоящее время определены 15, а остальные зарезервированы). Если данное поле имеет нулевое значение, то это говорит о передаче *полезной информации (Payload Data)*; в других случаях подпакет несёт служебные сведения.

Второе поле — восьмибитное поле *флагов*, его использование определяется типом подпакета.

Поле *длины* с разрядностью 16 бит заполняется суммарным значением длины подпакета с учётом полей заголовка.

Управляющие блоки включают различные параметры и флаги, зависящие от типа блока. Подпакеты *данных (DATA)* включают флаг управления сегментацией и сборкой, а также параметры *TSN*, *Stream ID*, *Stream Sequence Number* и *Payload Protocol Identifier*.

Перед фрагментом DATA размещаются *номер транспортной последовательности (Transport Sequence Number, TSN)*, *идентификатор потока*, *номер последовательности потока (Stream Sequence Number, SSN)*.

Номер транспортной последовательности используется для обеспечения надёжности каждой ассоциации, а номер последовательности потока — для упорядочивания по потокам. Отдельные сообщения в потоке отмечаются идентификатором потока.

Информационная часть предназначена для передачи собственно данных, которые определяются типом подпакета. Согласно протоколу SCTP, размерность подпакета должна быть кратна 32 битам. В противном случае информационная часть дополняется нулевыми значениями, но в поле длины указывается истинная величина. Это позволяет на приёмной стороне соединения исключить добавленные нули из передаваемых данных.

Параметр *Payload Protocol ID* включён для обеспечения возможности расширения в новых версиях протокола. Если предположить, что функции идентификации протокола и мультиплексирования по портам в будущем перестанут играть столь важную роль, как сейчас, *Payload Protocol ID* будет обеспечивать идентификацию протоколов, передаваемых с помощью SCTP без использования номера порта.

Формат сообщений SCTP обеспечивает механизм связывания множества блоков данных и управления в одно сообщение для повышения эффективности транспорта. Использованием такой *группировки (Bundling)* управляет приложение, поэтому группировка стартовой передачи невозможна. Связывание естественным образом осуществляется при повторе передачи блоков *DATA* в целях снижения вероятности насыщения.

6.3.2. Функции SCTP

SCTP представляет собой unicast-протокол, который обеспечивает обмен данными между двумя конечными точками.

Аналогом TCP-соединения для SCTP является ассоциация, которая устанавливается между двумя оконечными устройствами. При этом одно устройство может быть определено несколькими IP-адресами, список которых передаётся при установлении ассоциации. Для передачи данных через ассоциацию используются все возможные комбинации адресов пары оконечных устройств.

Отказоустойчивость в таком случае обеспечивается за счёт того, что разные IP-адреса присваиваются различным интерфейсам устройств, и трафик между ними передаётся по разным маршрутам. В случае отказа какого-либо оборудования в сети и недоступности одного или нескольких IP-адресов трафик продолжает передаваться между оставшимися адресами, и разрыва SCTP-ассоциации не происходит.

Описанный выше механизм работы SCTP-ассоциации носит название *многодомности (SCTP Multi-Homing)*.

К другим ключевым функциям протокола SCTP относятся:

— группировка различных сигнальных сообщений в одном пакете с одним SCTP/IP-заголовком (Chunk Bundling), что повышает эффективность использования полосы пропускания;
— последовательная доставка сообщений внутри различных потоков, что позволяет избежать ситуации, встречающейся при использовании протокола TCP, когда в случае потери одного пакета остальные задерживаются в буфере до успешной его перепосылки (Head-of-Line Blocking);
— использование контрольных сумм для обеспечения безошибочной передачи пакетов, а также для защиты от атак.

Протокол SCTP поддерживает ряд функций, унаследованных не только от TCP, но и от других протоколов. При этом в нём реализованы и дополнительные функции:

- *Сохранение границ сообщений.* Сообщения, передаваемые SCTP, размещаются в подпакетах (или фрагментах), что даёт возможность приложениям отделить одно сообщение от другого.
- *Отсутствие блокировок типа head-of-line.* В отличие от TCP протокол SCTP не требует строгой упорядоченности передаваемых пакетов. Поэтому в нём отсутствует задержка, вызываемая блокировкой обслуживания, возникающей при восстановлении TCP корректной последовательности пакетов.
- *Несколько режимов доставки.* SCTP может передавать данные как в строгом порядке (как TCP), так и частично упорядоченные (по потокам) и неупорядоченные вовсе (как UDP).
- *Поддержка многодомности.* SCTP может переадресовывать пакеты на альтернативный IP-адрес.
- *Контроль перегрузки*. SCTP использует стандартные методики, применяющиеся для контроля перегрузки в TCP, в том числе медленный старт, предотвращение перегрузки и быструю повторную передачу.
- *Выборочные подтверждения.* SCTP использует схему выборочного подтверждения, унаследованную из TCP, для восстановления утраченных пакетов.
- *Фрагментация пользовательских данных.* SCTP разбивает сообщения на фрагменты, чтобы *максимальный размер передаваемого элемента (Maximum Transfer Unit, MTU)* соответствовал ограничениям конкретного маршрута пересылки между взаимодействующими хостами (RFC 1191 [34]).
- *Механизм контроля работоспособности (Hearbeat)*. SCTP посылает пакеты контроля работоспособности на адреса находящегося в режиме ожидания хоста, которые входят в ассоциацию. Протокол декларирует, что IP-адрес будет отключён, как только он достигнет порогового значения невозвращённых подтверждений о работоспособности.
- *Защита от DoS-атак.* SCTP использует механизм cookie при инициализации ассоциации, чтобы смягчить воздействие DoS-атак.

6.3.3. Множественность потоков и варианты доставки

Название протокола SCTP обусловлено его многопотоковой природой передачи данных. Поддержка множества одновременных потоков позволяет распределить между этими потоками передаваемую информацию так, чтобы каждый из потоков обеспечивал независимую упорядоченную доставку данных. Потеря сообщения в любом из потоков оказывает влияние лишь на данный поток, не затрагивая работу других потоков данных.

Протокол TCP работает с одним потоком данных и обеспечивает сохранение порядка доставки байт из потока. Такой подход удобен для доставки файлов или записей, но он может приводить к дополнительным задержкам при потере информации в сети или нарушении порядка доставки пакетов. При возникновении подобных ситуаций протокол TCP должен дождаться доставки всех данных, требуемых для восстановления порядка.

В рамках одного соединения SCTP обеспечивает единый механизм управления потоком и контроля насыщения, что существенно снижает нагрузку на транспортный уровень.

SCTP разделяет понятия надёжной и упорядоченной доставки, в то время как в TCP эти два аспекта неразрывно связаны, так как все данные надёжно доставляются хосту-получателю и предоставляются приложению в той последовательности, в какой они передавались. Для этого TCP использует номер последовательности в заголовке каждого пакета.

Протокол SCTP поддерживает многопотоковую передачу за счёт устранения зависимости между передачей и доставкой данных. В частности, каждый блок полезной информации типа DATA (данные) использует два набора порядковых номеров. Номер TSN управляет передачей сообщений и детектированием их потери, а пара *идентификатор потока Stream ID–номер SSN* используется для управления порядком доставки потребителю полученных данных.

Такая независимость механизмов нумерации позволяет получателю незамедлительно обнаруживать пропуски данных, а также видеть влияние потерянных данных на поток. Утрата сообщения вызывает появление пропуска в порядковых номерах SSN для потока, на который это сообщение оказывает влияние и не вызывает такого пропуска для других потоков. Следовательно, получатель может продолжить доставку незатронутых потоков, не дожидаясь повтора передачи утраченного сообщения.

6.3.4. Многодомность

Механизм многодомности предназначен для повышения устойчивости сети к выходам из строя интерфейсов на хосте и ускорения восстановления в случае сбоя в сети. Но эффективность этого механизма падает, если путь взаимодействия внутри ассоциации проходит через единую точку сбоя сети.

Действующий вариант SCTP не поддерживает *распределения нагрузки (Load Sharing)*, поэтому многодомные хосты обеспечивают лишь избыточность соединений для повышения уровня надёжности. Один из адресов многодомного хоста указывается в качестве *основного (Primary)* и используется как адрес получателя для всех блоков данных при нормальной передаче. При передаче повторных блоков данных используется один из дополнительных адресов с целью повышения вероятности доставки в конечную точку. При повторяющихся неоднократно повторах передачи принимается решение об отправке всех блоков данных с использованием альтернативного адреса, пока системе мониторинга не удастся увидеть доступность основного адреса.

Для поддержки множества интерфейсов конечные точки SCTP обмениваются списками своих адресов в процессе создания ассоциации. Каждая из конечных точек должна быть способна принимать сообщения с любого адреса, связанного с удалённым партнёром; на практике некоторые операционные системы могут использовать в пакетах циклический перебор адресов отправителя, и в таких случаях приём пакетов с различных адресов является нормальной ситуацией. Для всего списка адресов конечной точки в данной сессии используется один номер порта.

Для повышения уровня безопасности требуется, чтобы некоторые отклики передавались по адресу, указанному в поле отправителя сообщения, вызвавшего отклик. Например, когда сервер получает блок INIT от клиента для инициирования SCTP-ассоциации, сервер всегда будет передавать блок INIT ACK по адресу отправителя в заголовке IP-блока INIT.

6.3.5. Установление ассоциаций

SCTP, как и TCP, ориентирован на установление соединения. Оба протокола требуют определения состояния соединения на каждом хосте. Соединение TCP определяется двумя IP-адресами и двумя номерами портов. Ассоциация SCTP определяется как набор IP-адресов + порт на каждом хосте. Любые из IP-адресов на любом хосте могут указываться в качестве отправителя или получателя в IP-пакете и это корректно идентифицирует ассоциацию.

Перед началом обмена данными два SCTP-хоста должны передать друг другу информацию о состоянии соединений с помощью четырёхэтапной процедуры установки соединения (handshake) (рис. 6.12).

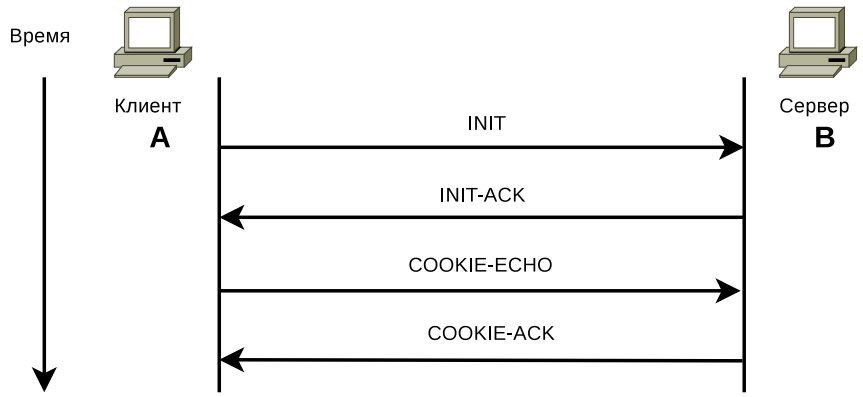


Рис. 6.12. Четырёхэтапная процедура установки соединения SCTP

Процедура, предусмотренная протоколом SCTP, позволяет защититься от DoS-атак. Получателю сообщения о намерении установить контакт *INIT* в четырёхэтапной процедуре установки соединения не требуется сохранять никакую информацию о состоянии или резервировать какие-либо ресурсы. Вместо этого он посылает в ответ сообщение *INIT-ACK*, которое включает в себя *запись состояния (Cookie)*, содержащую всю информацию, необходимую отправителю *INIT-ACK* для того, чтобы сформировать своё состояние. Запись состояния подписывается цифровой подписью. Оба сообщения, *INIT* и *INIT-ACK*, содержат несколько параметров, необходимых для установки начального состояния:

- список всех IP-адресов, которые станут частью ассоциации;
- номер транспортной последовательности, используемый для надёжной передачи данных;
- тег инициации, который должен быть включён в каждый входящий пакет SCTP;
- число выходящих потоков, запрашиваемых каждой из сторон;
- число входящих потоков, которые способна поддерживать каждая из сторон.

После обмена этими сообщениями, отправитель INIT возвращает назад запись состояния в виде сообщения COOKIE-ECHO, которое также может содержать связанные с ним пользовательские сообщения DATA. При получении COOKIE-ECHO получатель полностью меняет своё состояние и отправляет обратное сообщение COOKIE-ACK, подтверждающее завершение настройки. COOKIE-ACK также может сопровождаться пользовательскими сообщениями DATA.

6.3.6. Завершение работы ассоциации

Транспортному протоколу, ориентированному на соединение, необходим метод постепенного отключения ассоциации. SCTP использует процедуру установки соединения, отличающуюся от процедуры, применяемой в TCP: конечная точка TCP может инициировать процедуру отключения, сохраняя открытым соединение и получая новые данные от другого хоста. SCTP не поддерживает такого наполовину закрытого состояния, т.е. обе стороны не могут передавать новые данные на свой более высокий уровень, если инициирована последовательность постепенного отключения.

Пусть приложение на хосте A хочет отключить и закрыть ассоциацию с хостом B. SCTP устанавливает состояние SHUTDOWN_PENDING, в котором он не будет принимать данные от приложения, но по-прежнему будет посылать новые данные, помещаемые в очередь на передачу на хост B. После подтверждения всех размещённых в очереди данных хост A посылает подпакет SHUTDOWN и устанавливает состояние SHUTDOWN_SENT.

До получения подпакета SHUTDOWN хост B уведомляет свой более высокий уровень, что прекращает принимать от него новые данные и вводит состояние SHUTDOWN_RECEIVED. Хост B передаёт оставшиеся данные на A, за которыми следуют фрагменты SHUTDOWN, информирующие B о появлении данных и подтверждающие, что ассоциация отключена. Как только подтверждены все данные, помещённые в очередь на хосте B, хост A посылает соответствующий фрагмент SHUTDOWN-ACK, за которым следует фрагмент SHUTDOWN-COMPLETE, завершающий отключение ассоциации.

6.4. Протокол DCCP

Протокол DCCP (Datagram Congestion Control Protocol) [35,36] является транспортным протоколом, который использует двунаправленные уникастные соединения с управлением перегрузкой для ненадёжной доставки дейтаграмм.

Протокол DCCP имеет встроенную систему управления перегрузкой, включающую поддержку *уведомления о перегрузке канала (Explicit Congestion Notification, ECN)* [37] для ненадёжных потоков дейтаграмм, исключая непредсказуемые задержки, характерные для TCP, что обеспечивает надёжное согласование параметров при установлении соединения.

6.4.1. Характеристики DCCP

Протокол DCCP обладает следующими характеристиками:

- является протоколом для потоков пакетов, а не потоков байт;
- реализует поток дейтаграмм с подтверждением получения, но без повторной посылки;
- имеет ненадёжный диалог установления и разрыва соединения;
- обеспечивает надёжное согласование параметров;
- предоставляет выбор механизмов подавления перегрузки;
- является протоколом управления перегрузкой, а не протоколом управления потоками;
- имеет опции, указывающие отправителю, был ли пакет доставлен получателю, помечен ECN, повреждён или отброшен входным буфером получателя;

— осуществляет управление перегрузкой со встроенной индикацией явной перегрузки ECN;
— обладает механизмами, позволяющими серверу избежать поддержки состояний неподтверждённых попыток соединений;
— выявляет MTU пути.

6.4.2. Типы сообщений DCCP

Протокол DCCP использует девять различных типов сообщений:
— DCCP-Request инициирует соединение;
— DCCP-Response является ответом на запрос DCCP-Request;
— DCCP-Data передаёт данные;
— DCCP-Ack передаёт подтверждения о получении пакетов;
— DCCP-DataAck передаёт данные в сочетании с подтверждениями;
— DCCP-CloseReq запрашивает закрытие соединения;
— DCCP-Close осуществляет закрытие соединения или запускает процедуру сброса соединения (DCCP-Reset);
— DCCP-Reset осуществляет процедуру сброса соединения;
— DCCP-Sync, DCCP-SyncAck осуществляют повторную синхронизацию номеров пакетов после длительного периода потерь.

6.4.3. Формат заголовка DCCP

Базовый заголовок DCCP имеет следующий формат (рис. 6.13).

Рис. 6.13. Формат базового заголовка DCCP

Поля *Порт отправителя (Source Port)* и *Порт получателя (Dest Port)* (длиной по 16 бит каждый) идентифицируют соединение. Когда соединение формируется, клиент должен выбрать порт отправителя случайным образом, чтобы уменьшить вероятность атаки.

Поле *Смещение данных (Data Offset)* (длина 8 бит) указывает смещение от начала заголовка пакета DCCP первого октета данных (выражается в 32-битных словах).

Поле *CCVal* (длина 4 бита) используется отправителем CCID.

Поле *Checksum Coverage* (CsCov) (длина 4 бита) определяет части пакета, которые покрываются полем *Контрольная сумма*.

Поле *Контрольная сумма (Checksum)* (длина 16 бит) содержит контрольную сумму заголовка пакета DCCP (включая опции), псевдозаголовка сетевого уровня и, в зависимости от CsCov, полей данных приложений.

Поле *Зарезервировано (Reserved)* (длина 3 бита) содержит нули, получатель должен это поле игнорировать.

Поле *Тип (Type)* (4 бита) специфицирует тип пакета.

Поле *Расширенные порядковые номера (X)* (длина 1 бит) равно нулю, если передаются только младшие (LSB) 24 бита порядкового номера, а базовый заголовок имеет длину 12 байт и значение 1, если в заголовке используются 48-разрядные порядковые номера. Пакеты DCCP-Data, DCCP-DataAck и DCCP-Ack могут иметь значение, X равное 0 или 1. Все пакеты DCCP-Request, DCCP-Response, DCCP-CloseReq, DCCP-Close, DCCP-Reset, DCCP-Sync и DCCP-SyncAck должны иметь X=1.

Поле *Порядковый номер (Sequence Number)* (длина 48 или 24 бита) идентифицирует пакет в последовательности. Номер по порядку увеличивается на 1 после посылки каждого пакета, включая пакеты DCCP-Ack, которые не несут в себе данных.

После базового заголовка следует заголовок пересылаемого типа пакета.

6.4.4. Процедура взаимодействия

Процедура взаимодействия двух элементов следующая.

1) Клиент посылает серверу запрос DCCP-Request на установление соединения. Определяются номера портов клиента и сервера, запрашиваемая услуга и другие параметры соединения, включая CCID, необходимый серверу при работе с клиентом.
2) В ответ сервер посылает пакет-отклик.
3) Клиент посылает серверу подтверждение DCCP-Ack получения DCCP-отклика.
4) Далее по необходимости происходит обмен подтверждениями DCCP-Ack для согласования используемых параметров.
5) Сервер и клиент обмениваются пакетами DCCP-Data, DCCP-Ack.
6) Для закрытия соединения сервер посылает DCCP-CloseReq.
7) Для подтверждения закрытия соединения клиент посылает DCCP-Close.
8) Сервер посылает пакет DCCP-Reset, при этом состояние соединения ликвидируется.
9) Клиент получает пакет DCCP-Reset и сохраняет своё состояние в течение некоторого времени для завершения происходящих обменов.

6.4.5. Функциональность DCCP

Протокол DCCP может реализовать механизм контроля за перегрузкой, многодомность и мобильность (за счёт механизма переадресации), процедуру медленного получателя (Slow Receiver). DCCP не предоставляет криптографических гарантий безопасности, но имеет возможности противостоять некоторым видам атак благодаря используемой системе нумерации пакетов.

Глава 7. Протоколы верхних уровней

7.1. Служба доменных имён

Хосты[1] адресуются с помощью IP-адресов. Но для человека обращение по цифровому адресу достаточно затруднительно. Для преодоления этой проблемы предложена служба, преобразующая IP-адреса в имена хост-машин, — *служба доменных имён (Domain Name System, DNS)*.

На раннем этапе существования Интернета (ARPANET) рост этой сети был умеренным, и отображение имён хост-машин на IP-адреса поддерживалось *сетевым информационным центром (Network Information Center, NIC)* с помощью единственного файла (hosts.txt). Каждый администратор хост-машины или организации периодически копировал этот файл на каждую подсоединённую к сети хост-машину. Но по мере роста и изменения состава сети, из-за постоянного роста частоты выборки файла hosts.txt из NIC, необходимости отделения вопросов управления локальными именами и адресами в разных организациях, а также необходимости всё более частого внесения изменений в файл hosts.txt стало ясно, что централизованная схема оказывается неработоспособной, и должно быть найдено альтернативное решение.

Первая реализация DNS называлась JEEVES. Более поздней реализацией стала *BIND (Berkeley Internet Name Domain)*, написанная для 4.3BSD UNIX и являющаяся на сегодняшний день наиболее популярной реализацией DNS.

Первоначальные результаты разработки DNS были опубликованы в 1983 г. в RFC 882 [38] и RFC 883 [39]. После экспериментов с несколькими реализациями DNS была формально определена в RFC 1034 [40] и RFC 1035 [41] в 1987 г.

Основные концепции DNS:

— распределённая база данных, хранящая обобщённые записи о ресурсах сети (resource records), с децентрализованным управлением;
— схема именования основывается на иерархически структурированных доменных именах.

Корневое имя (Root's Name) иерархии DNS обозначается одиночной точкой («.»). Каждый *узел (Node)* дерева представляет раздел общей базы данных или *домен (Domain)*. Каждый домен в дальнейшем может делиться на подразделы, называемые в DNS *поддоменами*. Поддомены представляются как потомки своих родительских узлов (Parent Nodes). Каждый домен имеет *метку (Label)*, которая идентифицирует его местоположение относительно его родительского домена. Кроме того, домен имеет *доменное имя (Domain Name)*, которое идентифицирует его местоположение в базе данных DNS. Полное доменное имя представляет собой последовательность меток от корневого домена, которые разделяются между собой символом «.».

Каждый хост в сети имеет доменное имя, которое является указателем на информацию об этой хост-машине. Эта информация может содержать IP-адрес, маршрутную информацию почтовой системы и т. д. Хост может иметь одно или несколь-

[1] Слово «хост» не является в полном смысле синонимом имени компьютера, так как у компьютера может быть множество IP-адресов, каждому из которых можно поставить в соответствие одно или несколько доменных имён. Кроме того, одному доменному имени можно поставить в соответствие несколько разных IP-адресов, которые, в свою очередь, могут быть закреплены за разными компьютерами.

ко доменных *имён-псевдонимов (Domain Name Aliases)*, которые являются простыми указателями одного доменного имени (имени-псевдонима) на другое (*каноническое доменное имя — Canonical Domain Name*).

Именовать хост можно либо частичным именем, либо полным именем.

Полное имя хоста — это имя, в котором перечисляются слева направо имена всех промежуточных узлов между листом и корнем дерева доменного именования, при этом начинают с имени листа, а заканчивают корнем (например, www.sci.pfu.edu.ru.).

Частичное имя — это имя, в котором перечислены не все, а только часть имён узлов:

- www
- www.sci
- www.sci.pfu.edu

В частичных именах символ точки в конце имени не ставится. Программное обеспечение системы доменных имён расширяет неполные имена до полных, прежде чем обратиться к серверам доменных имён за IP-адресом.

Система доменных имён состоит из трёх основных частей (RFC 1034 [40], RFC 1035 [41]):

- всего множества доменных имён (Domain Name Space);
- серверов доменных имён (Domain Name Servers);
- клиентов DNS (Resolver).

7.1.1. Схемы разрешения имён

Обычно используют две основные схемы разрешения имён: *нерекурсивную* и *рекурсивную* (RFC 1034 [40] и RFC 1035 [41]).

Процедура нерекурсивного разрешения запроса выглядит следующим образом.

1. Прикладная программа запрашивает IP-адрес по доменному имени у местного сервера (запрос клиента рекурсивный, т.е. клиент просит сервер найти ему адрес).
2. Местный сервер сообщает прикладной программе IP-адрес запрошенного имени, выполняя при этом нерекурсивный опрос серверов доменных имён. При этом:
   а) если адрес находится в зоне ответственности местного сервера, сразу сообщает его клиенту;
   б) если адрес находится в зоне ответственности другого сервера доменных имён, то обращается к корневому серверу системы доменных имён за адресом TLD-сервера (Top-Level Domain Server);
   в) обращается к TLD-серверу за адресом;
   г) получает от него адрес удалённого сервера;
   д) обращается к удалённому серверу за адресом;
   е) получает от удалённого сервера адрес.

Собственно нерекурсивным рассмотренный выше запрос является только с точки зрения сервера. С точки зрения клиента процедура разрешения запроса является рекурсивной, так как клиент перепоручил локальному серверу доменных имён заниматься поиском необходимой информации.

Согласно RFC 1035 [41] клиент и сам может опрашивать удалённые серверы доменных имён и получать от них ответы на свои запросы. В этом случае клиент обращается к локальному серверу доменных имён. Если клиент не получает от него адреса, то опрашивает сервер корневого домена, получает от него адрес удалённого сервера TLD, опрашивает этот сервер, получает адрес удалённого сервера, опрашивает удалённый сервер и получает IP-адрес в случае посылки так называемого «прямого» запроса.

Если пользователь обращается в течение короткого времени к одному и тому же ресурсу сети, то запрос на удалённый сервер не отправляется, а информация ищется в кэше.

Вообще говоря, порядок обработки запросов можно описать следующим образом:

1) поиск ответа в локальном кэше;
2) поиск ответа на локальном сервере;
3) поиск информации в сети.

При этом кэш может быть как у клиента, так и у сервера.

Отличие рекурсивной процедуры от описанной выше нерекурсивной процедуры состоит в том, что удалённый сервер сам опрашивает свои серверы зон, а не сообщает их адреса местному серверу доменных имён. Основная нагрузка в этом случае ложится на местный сервер доменных имён, который осуществляет опрос всех остальных серверов. Для того чтобы сократить число таких обменов, если позволяет объём оперативной памяти, можно разрешить буферизацию (кэширование) адресов. В этом случае число обменов с удалёнными серверами сократится. При этом локальный сервер сразу получает от удалённого адрес хоста, а не адреса серверов поддоменов. Удалённому серверу при этом должно быть разрешено обслуживание рекурсивных запросов с соответствующего IP-адреса, местный сервер должен обратиться к удалённому с рекурсивным запросом.

7.1.2. Типы серверов доменных имён

RFC 1034 [40] и RFC 1035 [41] выделяют несколько типов DNS-серверов. В соответствии с типами откликов на запрос к системе доменных имён серверы можно разделить на *авторитативные (Authoritative)* и *неавторитативные (Non Authoritative)*.

Авторитативный отклик (Authoritative Response) возвращают серверы, которые являются ответственными за зону, в которой описана информация, необходимая клиенту DNS.

Неавторитативный отклик (Non Authoritative Response) возвращают серверы, которые не отвечают за зону, содержащую необходимую клиенту информацию.

Авторитативный отклик могут, в свою очередь, вернуть либо *master-сервер зоны (Primary Server)*, либо *slave-сервер (Secondary Server) зоны*. В русскоязычной литературе их называют *основным* и *дублирующим* серверами (или первичным и вторичным соответственно).

Master-сервер (Primary — первичный) доменных имён является ответственным за информацию о зоне в том смысле, что читает описание зоны с локального диска компьютера, на котором он функционирует, и отвечает в соответствии с этим описанием на запросы клиентов. Описание зоны master-сервера является первичным, так как его создаёт вручную администратор зоны. Соответственно

вносить изменения в описание зоны может только администратор данного сервера. Все остальные серверы только копируют информацию с master-сервера. Для зоны можно определить только один master-сервер, так как первоисточник может и должен быть только один.

Slave-сервер (Secondary — вторичный, дублирующий) также является ответственным за зону. Его основное назначение заключается в подстраховке работы основного сервера доменных имён, ответственного за зону, на случай его выхода из строя, а также в разгрузке основного сервера посредством перенаправления части запросов на себя.

Администратор slave-сервера не прописывает данные описания зоны, а лишь обеспечивает настройку своего сервера таким образом, чтобы тот копировал описание зоны с master-сервера, поддерживая его в актуальном согласованном с master-сервером состоянии.

Обычно время согласования описания зоны между slave-сервером и master-сервером задаётся администратором master-сервера в описании зоны. Slave-сервер в момент своего запуска копирует это описание и затем руководствуется им при обновлении информации о зоне. Slave-сервера периодически через заданный интервал времени опрашивают master-сервер на предмет изменения описания зоны. Если такие изменения есть, то описание зоны копируется на slave-сервер.

Спецификация DNS позволяет реализовать и другой механизм обновления информации — *оповещение об изменениях (DNS Notify)*. В этом случае инициатива обновления описаний зоны на slave-серверах принадлежит уже master-серверу. Последний оповещает slave-серверы о том, что в базу были внесены изменения и что необходимо эти изменения скопировать на slave-серверы.

Принцип работы DNS Notify следующий:

1) в базу данных первичного сервера вносятся изменения;
2) первичный сервер оповещает свои вторичные сервера о том, что произошли изменения, сообщая им номер версии описания зоны;
3) вторичный сервер запрашивает у первичного описание зоны и, если номера версии описаний зоны не совпадают (на первичном сервере номер больше), то инициирует процесс обновления описания зоны;
4) завершив обновление описания зоны, вторичный сервер посылает оповещения на известные ему авторитативные сервера зоны.

В настоящее время существует два механизма копирования зоны: *полное копирование (AXFR)* и *инкрементальное (incremental) копирование зоны (IXFR, RFC 1995 [42])*.

При традиционном обмене описанием зоны (AXFR) между первичным и вторичным серверами передаётся полное описание зоны.

Для того чтобы не передавать всю зону, а передавать только изменения, предназначен механизм инкрементальной передачи описания зоны (IXFR). В рамках обмена передаются номера версий описаний зон и записи, которые нужно добавить или удалить. Сначала идут номер старой версии и список записей, которые нужно удалить, а потом номер более свежей версии и записи, которые нужно добавить.

Существует оговорённая практика резервирования серверов, которая описана в рекомендациях по ведению зон. Она заключается в том, что для домена второго уровня необходимо иметь как минимум два сервера, ответственных за зону, т.е. дающих авторитативные отклики на запросы: один первичный сервер и один вторичный сервер. При этом эти серверы должны иметь независимые подключения к Интернету, чтобы обеспечить бесперебойное обслуживание запросов к

зоне в случае потери связи с одним из сегментов сети, в котором находится один из серверов.

В современных RFC, расширяющих толкование механизмов взаимодействия между участниками обмена данными в рамках DNS, типизацию серверов и их разделение на master-серверы и slave-серверы дают относительно процедур копирования зоны.

Вторичным сервером называют сервер, который использует механизм передачи зоны для получения копии зоны, а *первичным* называют сервер, с которого осуществляется копирование зоны. При этом зону можно скопировать с любого сервера, являющегося авторитативным. То есть зону можно скопировать и с вторичного сервера, который относительно самой процедуры копирования зоны будет считаться первичным сервером. Для того чтобы выделить сервер, который не копирует зоны ни с какого другого сервера, вводят понятие *первичного мастер-сервера (Primary Master)*. Этот сервер для зоны только один, и он находится в корне процедуры копирования описания зоны.

Типизация серверов относительно процедуры обмена описаниями зон связана с возможностями, которые не описаны в RFC 1034 [40] и RFC 1035 [41], и соответственно не поддерживались в ранних версиях программ-серверов доменных имён. Это *механизм объявлений об изменениях в описании зоны* (RFC 1996 [43]), *динамическое обновление описания зоны* (RFC 2136 [44]) и *инкрементальное обновление описания зоны* (RFC 1995 [42]).

Невидимые сервера (Stealth, RFC 1996 [43]) не упоминаются в описании зоны. Таким образом, их никто не видит, так как в рамках DNS-обмена данными информацию о них получить нельзя ни путём простых запросов, ни посредством копирования описания зоны. Тем не менее существуют ещё *файлы статической настройки (конфигурации) серверов доменных имён*, где такой сервер может быть прописан. Возможное применение таких серверов — внесение обновлений в зону, находящуюся за брандмауэром. В этом случае первичный мастер-сервер можно сделать невидимым, что позволяет нейтрализовать атаки на зону.

Другая причина создания невидимых серверов — разгрузка официально зарегистрированных серверов. В этом случае для обслуживания определённого класса клиентов, которых можно настроить на работу с невидимым сервером, создаётся один или несколько вторичных серверов. Они являются авторитативными, но неизвестными другим.

Выделяют ещё один тип серверов — *кэширующие (Cache) сервера*. Сервер данного типа не является авторитативным для какой-либо зоны. Такие серверы используют для организации централизованного кеширования соответствий доменных имён и IP-адресов. Идея организации кэширующего сервера состоит в том, чтобы не искать соответствия доменного имени и IP-адреса в сети, а накапливать их в своём локальном кэше и обслуживать оттуда запросы клиентов.

Отдельно можно выделить сервера, обслуживающие корневую зону (Root Servers). Их место в получении отклика на запрос к системе доменных имён ключевое. Именно к одному из корневых серверов обращается локальный сервер доменных имён, если не находит в зоне своей ответственности или в своём кэше соответствия между доменным именем и IP-адресом.

7.1.3. Динамическое обновление зоны

Изначально описание зоны было, да и до сих пор в большинстве случаев остаётся, статическим. Это значит, что есть на первичном сервере файл описания зо-

ны, в который администратор вручную или при помощи скриптов изменения содержания файла вносит изменения. Для того чтобы они стали актуальными, т.е. их увидели клиенты, необходима перезагрузка сервера. Идея *динамического обновления описания зоны (Dynamic Updates или DNS UPDATE)* состоит в том, чтобы, во-первых, вносить изменения в описание зоны без перезагрузки сервера, а, во-вторых, делать это удалённо, т.е. администратору не нужно получать доступ к файлам описания зон ни для ручного редактирования, ни для запуска скриптов.

Чаще всего необходимость динамического обновления связывают с работой по протоколу *DHCP (Dynamic Host Configuration Protocol, RFC 1541 [45])*, который позволяет динамически назначать компьютерам IP-адреса, маски, доменные имена и т.п., т.е. передавать хостам данные, без которых невозможна работа в сетях TCP/IP. Динамическое обновление позволяет авторизованным клиентам посылать запросы на динамическое обновление описания зоны серверам, которые являются авторитативными для соответствующей зоны. Запросы могут направляться как к первичным серверам, так и к вторичным. В рамках динамического обновления можно удалять и добавлять отдельные записи описания ресурсов в описания зоны, наборы записей описания ресурсов, выделенных по определённому признаку.

7.2. ENUM и E.164

7.2.1. ENUM — tElephone NUmber Mapping

Для интеграции систем доступа в ТфОП и IP-сетях предложен протокол *ENUM (tElephone NUmber Mapping)* [46] как попытка совместить телефонную нумерацию и IP-адресацию. ENUM (RFC 2916 [47]) — это сетевой протокол, определяющий выбор маршрутов для связи с различными устройствами, принадлежащими одному абоненту (пользователю телефонного номера в международном формате E.164). ENUM устанавливает соответствие между номером в формате E.164 (международный формат телефонных номеров, определяемый в Рекомендации E.164 ITU) и доменным именем (Domain Name System, DNS) [48].

7.2.2. Международный план нумерации E.164

Рекомендация МСЭ-Т E.164 «План нумерации для международной связи общего пользования» описывает структуру, компоненты и функциональные возможности четырёх категорий номеров, обеспечивающих маршрутизацию вызовов в сети общего пользования:

— для географических областей;
— для глобальных услуг;
— для сетей;
— для групп стран.

Рекомендуемое МСЭ-Т максимальное число цифр в номерах для географических, глобальных услуг, сетей и групп стран должно быть равно 15 (исключая международный префикс).

Для географических областей поле «код» страны (CC, Country Code) содержит от 1 до 3 цифр, а поля «национальный код пункта назначения» и «номер абонента» (NDC+SN, National Destination Code + Subscriber Number) — от 12 до 14 цифр. В номере для глобальных услуг CC — 3 цифры, а в поле «глобальный номер абонента» (GSN, Global Subscriber Number) — 12 цифр. Номер для сетей CC

содержит 3 цифры, поле «идентификационный код» (IC, Identification Code) — от 1 до 4 цифр, а поле SN — от 8 до 11 цифр. Номер для групп стран поле CC содержит 3 цифры, поле SN — 11 цифр, а поле «идентификационного кода группы» (GIC, Group Identification Code) — 1 цифру.

7.2.3. Особенности структуры протокола ENUM

Документ IETF RFC 2916 [47] рассматривает процедуру использования системы доменных имён для хранения номеров E.164 и описывает процесс идентификации услуги по заданному номеру E.164. Вопросы маршрутизации соединения, используемого услугой, в RFC 2916 не обсуждаются. С помощью преобразования номеров E.164 в доменные имена и использования возможностей DNS, таких как *делегирование полномочий и записи NAPTR (Naming Authority Pointer)*, определяется доступность услуги для заданного доменного имени.

7.2.3.1. Номер E.164 в DNS

Домен «e164.arpa.» в DNS предоставляет инфраструктуру для хранения номеров E.164. Для упрощения распределённых операций доменная область разделяется на поддомены. Владельцы номеров E.164 для внесения их в DNS обращаются к администратору зоны, после чего осуществляется *проверка записи ресурса SOA (Start of Authority)*, ассоциированной с данной зоной, подобно обычным операциям в DNS.

Для определения имени DNS для заданного номера E.164 необходимо выполнить процедуру, состоящую из следующих шагов:

1) проанализировать полную форму записи номера E.164, включая код страны, например: +7-095-3689155;
2) удалить все символы, за исключением «+»: +70953689155;
3) удалить все символы, за исключением цифр: 70953689155;
4) установить точки («.») между каждой цифрой: 7.0.9.5.3.6.8.9.1.5.5;
5) установить обратный порядок цифр: 5.5.1.9.8.6.3.5.9.0.7;
6) добавить «.e164.arpa.» в конце строки: 5.5.1.9.8.6.3.5.9.0.7.e164.arpa.

Символ «+» сохраняется на этапе 2 для индикации принадлежности данного номера плану нумерации E.164.

7.2.3.2. Выбор универсального идентификатора ресурса URI, определяющего номер E.164

Запись указателя NAPTR в DNS используется для определения возможных путей обращения к специфическому ресурсу, идентифицируемому заданным именем. Запись используется также при определении услуг, существующих для специфического доменного имени, включая телефонные номера с использованием домена e164.arpa.

В процессе определения адреса URI используется запись ресурса *NAPTR RR (NAPTR Resource Record)*:

— поле *Order* определяет порядок, в котором обрабатываются записи при возвращении нескольких записей NAPTR в ответ на один запрос;
— поле *Preference* определяет порядок обработки записей при одном и том же значении поля Order нескольких записей NAPTR;

— поле *Service* определяет протокол и услуги, доступные при применении процедуры перезаписи, определённой в полях Regexp или Replacement;
— поле *Flags* содержит модификаторы, корректирующие процесс следующего поиска в DNS, и обычно применяются для оптимизации;
— поле *Regexp* вместе с полем Replacement используются для индикации правил перезаписи.

Процедуры подстановки и поиска выполняются на клиентских компьютерах, а не на серверах DNS. Идентификаторы URI хранятся в полях Regexp.

7.2.3.3. Процедура E2U

Процедура E2U для отображения номера E.164 в URI использует следующие данные:

— наименование процедуры: E.164 в URI;
— мнемоническое наименование: E2U;
— количество операндов: 1;
— тип операнда: номер E.164;
— формат операнда: первый операнд является номером E.164 в форме, определяемой шагом 2 алгоритма преобразования;
— алгоритм: непрозрачный;
— выходные данные: один или более URI;
— условия возникновения ошибок:
  – номер E.164 не входит в план нумерации;
  – номер E.164 входит в план нумерации, но для него не существуют URI;
  – услуга недоступна.
— аспекты безопасности:
  – преднамеренная переадресация;
  – отказ в услуге (удалив соответствующий URI номера E.164, злоумышленник может лишить клиента возможности доступа к ресурсу).

7.2.3.4. Уровневая архитектура ENUM

Корневой сервер имён может содержать лишь поддомены, соответствующие определённым МСЭ-Т кодам стран. Государство–член МСЭ-Т осуществляет управление доменом ENUM и планом нумерации E.164, соответствующим коду страны. При выделении номеров E.164 и соответствующих доменов ENUM должно быть обеспечено их взаимно-однозначное соответствие. Заполнение записи ENUM разрешается лишь для пользователя с выделенным номером E.164 (или для его агента). Когда использование номера прекращается, права на сохранение записей ENUM аннулируются, а сами записи уничтожаются.

Реестров уровня 1 для кода CC может быть несколько, например, интегрированный план нумерации может быть разделён между национальными Администрациями связи, а реестр уровня 1 может быть разбит на диапазоны номеров внутри CC. Для каждого номера E.164 может существовать не более одного реестра. Реестры назначаются соответствующим государством–членом МСЭ-Т [49].

Государство, входящее в МСЭ-Т и желающее включить свои ресурсы нумерации в домен e164.arpa и участвовать в системе ENUM, должно обеспечить реализацию ряда положений, а именно:

1) Государство при включении своих ресурсов нумерации в ENUM назначает реестры, связанные с этими ресурсами. Государство выбирает один реестр или представляет различные сегменты своего кода страны в различных реестрах. Выбор одного реестра упрощает многоуровневое администрирование.
2) Государство выбирает процедуру заполнения реестра уровня 1 номерами. Для обеспечения переносимости номера может потребоваться делегирование от реестра на индивидуальную основу номера. Такое случается, если не все номера в блоке, например, центральный код офиса, обязательно связаны с одним провайдером услуг.
3) Государство определяет правила для объектов, которые могут выполнять функцию регистратора услуги для своих членов. В общем случае любой объект может исполнять роль регистратора услуги. В альтернативном варианте провайдер услуг телефонии должен выполнять функции регистратора услуги для данного номера E.164.
4) Как отмечалось ранее, главным вопросом в административной модели для ENUM является проверка полномочий владельцев номеров по заполнению реестров своими записями для соответствующих номеров. Поэтому необходимо предусмотреть меры упрощения проверки достоверности выделения номеров и идентификации запрашиваемых записей ENUM.
5) Провайдеры услуг телефонии могут иметь необходимость поддерживать записи ENUM для номеров, которым они предоставляют услуги, отдельно от записей, поддерживаемых полномочным владельцем номера. Так как все записи ENUM для заданного номера должны храниться на одном и том же именном сервере, в административной модели необходимо определить статус провайдеров услуг телефонии (специальный или обычный, как у любого другого провайдера прикладных услуг).
6) Следует добавить ряд возможных функций ENUM по обеспечению информационной безопасности, а именно, в части
   - Безопасности информации в DNS. При чтении данных DNS, хранящихся в системе ENUM, должны быть предоставлены гарантии достоверности информации. Если клиенты имеют право добавлять, изменять или удалять записи из системы ENUM, то они должны быть уверены, что
     а) обновление данных осуществляется в директории услуги;
     б) они получат постоянный доступ к данным в требуемом объёме;
     в) обновление данных разрешено лишь им;
   - Заимствования прав. Имитация соединения (спуфинг) или неверное представление субъекта источника информации может разрешить неавторизованное обновление базы данных. Неправильные или недостающие данные могут в свою очередь вызвать преднамеренную переадресацию услуги или отказ в её предоставлении. Следовательно, клиенты, желающие обновить или добавить записи в службе ENUM, должны иметь возможность их однозначной идентификации для системы DNS.
   - Преднамеренного изменения данных. В процессе передачи записей ENUM, идентификаторы URI могут быть заменены, что, в свою очередь, может вызвать преднамеренную переадресацию.

Глава 8. QoS и передача мультимедийных данных

8.1. Базовые понятия QoS

Качество обслуживания (Quality of Service, QoS) определяется как мера производительности передающей системы, отражающая качество передачи и доступность услуг.

Качество передачи определяется следующими факторами:

- *доступность (Availability)*:
  - *сетевая доступность* — диапазон времени сетевой достижимости между входной и выходной точкой сети;
  - *доступность сервиса (Service Availability)* — диапазон времени, в течение которого этот сервис доступен между определёнными входной и выходной точками с параметрами, оговорёнными в *соглашении об уровне обслуживании (Service Level Agreement — SLA)*;
- *потери пакетов (Packet Loss)* — отношение правильно принятых пакетов к общему количеству пакетов, которые были переданы по сети;
- *задержка (Delay)* — время, которое требуется пакету для того, чтобы после передачи дойти до пункта назначения:
  - *задержка сериализации (Serialization Delay)*[1] — время, которое требуется устройству для передачи пакета заданного размера при заданной ширине полосы пропускания;
  - *задержка распространения (Propagation Delay)* — время, которое требуется переданной в канал единице информации для достижения принимающего устройства (зависит от расстояния и среды передачи);
  - *задержка коммутации (Switchiing Delay)* — время, которое требуется устройству, принявшему пакет, для начала передачи его следующему устройству;
- *колебания задержки (Packet Jitter)* — разница между сквозным временем задержки, которая возникает при передаче по сети разных пакетов[2];
- *пропускная способность (Bandwidth)* — общее количество данных, которые могут быть переданы в единицу времени между двумя точками присутствия оператора.

В пакетных сетях в информационном потоке может передаваться разнородный трафик, характеризующийся критичными и второстепенными для себя параметрами. Для передачи аудио- и видеоданных требуются разные требования к QoS. Для передачи видеоданных необходима высокая пропускная способность и стабильное время задержки при передаче. При этом, чтобы избежать искажений изображения, необходим стационарный поток данных. При интерактивной передаче звука требуется меньше пропускной способности канала, чем при передаче видео, но необходима малая задержка прохождения пакетов через сеть, иначе возникает «эхо». Передача файлов требует высокой пропускной способности, но, в

[1] Задержку сериализации называют ещё *задержкой передачи (Transmission Delay)*.

[2] Так, например, если для передачи одного пакета по сети требуется 100 мсек, а для передачи следующего пакета — 125 мсек, то колебание задержки составит 25 мсек.

отличие от большинства других видов сетевого трафика, наименее чувствительна к длительным и непостоянным задержкам в сети.

Качество обслуживания использует распределение по категориям и назначение приоритетов трафикам, что позволяет гарантировать трафику с большим приоритетом лучшие условия передачи через сетевую магистраль, вне зависимости от требований к пропускной способности трафика менее важных приложений.

8.2. Механизмы обеспечения QoS

В настоящее время существует целый ряд технологий, способных обеспечить качество обслуживания в сетях:

- обеспечение перекрывающей пропускной способности;
- установление приоритетов в виртуальных сетях;
- технология IntServ и протокол RSVP;
- технология DiffServ;
- организация приоритетных очередей в маршрутизаторах.

8.2.1. Обеспечение перекрывающей пропускной способности

Использование высокоскоростных каналов связи, предоставляемых, например, технологиями Fast/Gigabit Ethernet, при достаточно низкой загрузке сети позволяет избежать возникновения узких мест в сети. Низкая задержка и небольшая амплитуда дрожания достигаются за счёт отказа от маршрутизации и других методов, способных вызвать потерю пакетов и их повторную передачу. Однако в подавляющем большинстве случаев всё же необходим жёсткий контроль за распределением трафика.

8.2.2. Установление приоритетов в виртуальных сетях

Комитетом IEEE 802 разработаны стандарты IEEE 802.1Q и IEEE 802.1p, которые должны обеспечить взаимодействие виртуальных сетей и гарантировать пользователям необходимое качество обслуживания на основе присвоения приоритета. Механизм присвоения приоритета основан на указании приоритета передаваемого кадра. Этот механизм описан в стандарте IEEE 802.1p. Новые поля в кадре, которые служат для указания приоритета, регламентированы стандартом IEEE 802.1Q.

К кадру Ethernet добавлены два байта. Они определяют принадлежность кадра к определённой виртуальной сети и его приоритет. Можно задать до восьми уровней приоритета, благодаря чему происходит распределение кадров по очередям коммутатора. Этот механизм позволяет без задержек обрабатывать чувствительный к дрожанию трафик. Но поскольку отсутствует механизм контроля за действием пользователей по запросу на приоритет, сетевой администратор должен осуществлять контроль за поведением трафика.

8.2.3. Алгоритмы обработки трафика

8.2.3.1. Алгоритмы, использующиеся при распределении трафика по классам

8.2.3.1.1. Time Sliding Window with Two Color Marking. Алгоритм *скользящего временного окна с 2 цветным маркером — (Time Sliding Window with Two Color Marking, TSW2CM)* (рис. 8.1) имеет 2 составляющие: оценщик интенсивности и маркировщик, приписывающий каждому пакету определённый цвет в дополнение к приоритету сброса.

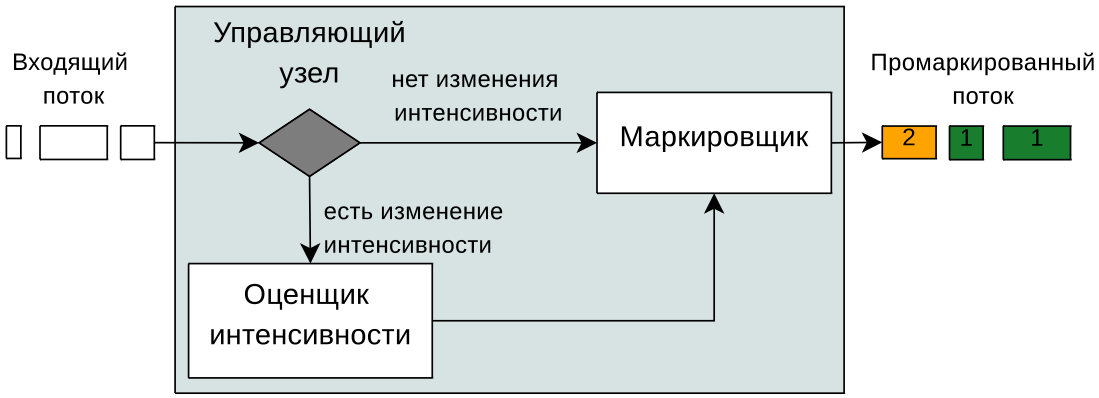


Рис. 8.1. Схема работы TSW2CM

При поступлении пакета оценщик оценивает скорость поступления информации с учётом всплесков интенсивности поступления трафика, строит оценку, аппроксимирующую долгосрочную измеренную интенсивность входящего потока. Маркировщик использует оценённую интенсивность для того, чтобы приписать пакету один из двух цветов, определяющих дополнительный приоритет.

8.2.3.1.2. A Single Rate Three Color Marker. *Алгоритм трёхцветного маркера для одного входящего потока (A Single Rate Three Color Marker, srTSM)* работает по аналогичному TSW2CM принципу.

Измеритель (Meter) измеряет пакеты и передаёт результаты измерения маркировщику (Marker). Маркировщик srTCM измеряет пакеты входящего IP-потока и маркирует пакеты зелёным, жёлтым или красным цветом. Маркировка основывается на параметре *Committed Information Rate (CIR)* и двух параметрах, ассоциированных с взрывным трафиком — *Committed Burst Size (CBS)* и *Excess Burst Size (EBS)*. Пакет маркируется зелёным цветом, если он не превосходит CBS, жёлтым, если интенсивность поступления превосходит CBS, но не превосходит EBS, и красным в ином случае.

8.2.3.1.3. A Two Rate Three Color Marker. *Двухпараметровый трёхцветный маркер (Two Rate Three Color Marker (trTCM))* измеряет пакеты IP потока и маркирует пакеты зелёным, жёлтым или красным цветом. Пакет маркируется красным, если размер пакета превосходит *Peak Information Rate (PIR)*. Иначе поток окрашивается либо жёлтым, либо зелёным цветом, в зависимости от того, превышено или нет значение параметра *Committed Information Rate (CIR)*.

Конфигурация trTCM осуществляется путём установки режима работы и заданием значений четырёх параметров трафика: *Peak Information Rate (PIR)* и связанного с ним *Peak Burst Size (PBS)*, *Committed Information Rate (CIR)* и связанного с ним параметра *Committed Burst Size (CBS)*.

Алгоритм trTCM используется для маркировки потока IP-пакетов в услугах, в которых различные уровни обслуживания соответствуют разным цветам. Например, услуга может сбрасывать все красные пакеты, продвигать вперёд пакеты жёлтого цвета и зелёные пакеты с низкой вероятностью сброса.

8.2.3.2. Алгоритм формирования трафика

8.2.3.2.1. Token Bucket Filter. *Token Bucket Filter (TBF) или «маркерное ведро»* — простая дисциплина очереди, которая передаёт поступающие пакеты со скоростью, не превышающей административно заданный порог, но с возможностью превышающих его коротких всплесков (рис. 8.2).

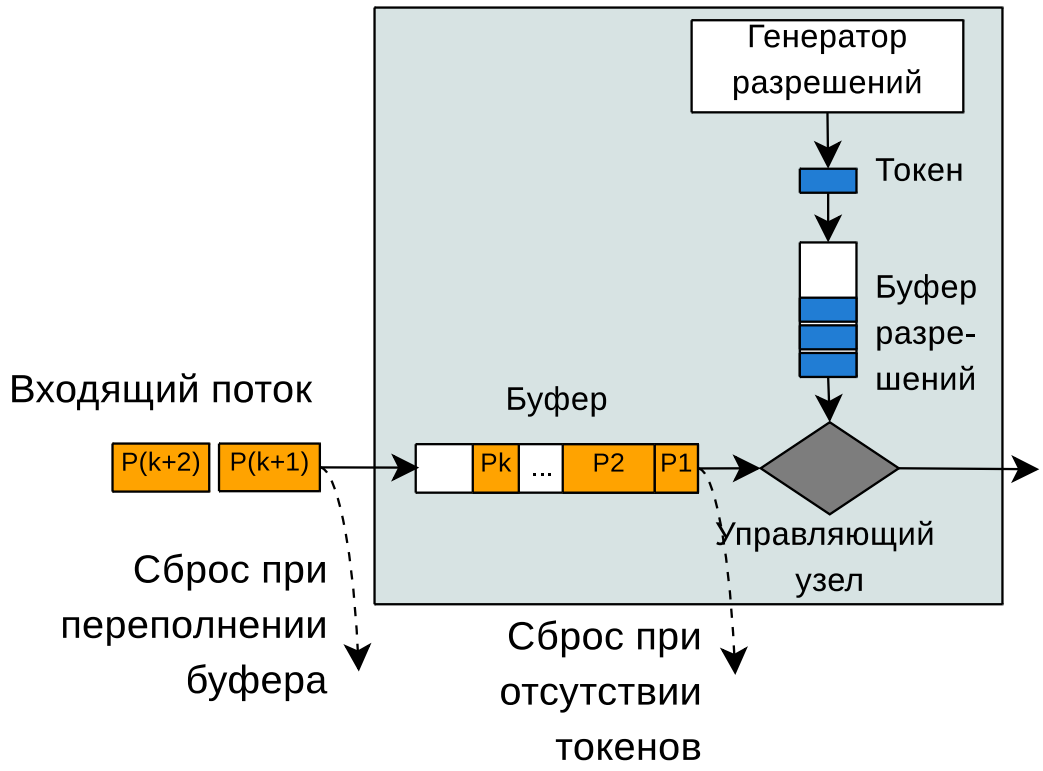


Рис. 8.2. Дисциплина Token Bucket Filter

Максимальная средняя скорость отправки потока пакетов из управляющего узла зависит от скорости прибытия в него разрешений на передачу N единиц данных. Очередной пакет может быть отправлен только при получении числа разрешений, достаточного для передачи данных, объём которых больше или равен размеру пакета. Если пакет поступит в управляющее устройство, не располагающее необходимым количеством разрешений, он будет отброшен также как и пакет, поступивший в переполненный буфер-формирователь.

Реализована TBF в виде буфера, постоянно заполняющегося токенами с заданной скоростью. Наиболее важным параметром буфера является его размер,

определяющий количество хранимых токенов.

Если данные прибывают со скоростью, равной скорости входящих токенов, то каждый пакет имеет соответствующий токен и проходит очередь без задержки.

Если данные прибывают со скоростью, меньшей скорости поступления токенов, то лишь часть существующих токенов будет уничтожаться, поэтому они станут накапливаться до размера буфера. Далее накопленные токены могут использоваться при всплесках, для передачи данных со скоростью, превышающей скорость пребывающих токенов.

Если данные прибывают быстрее, чем токены, то в буфере со временем не останется токенов, что заставит дисциплину приостановить передачу данных. Эта ситуация называется «превышением». Если пакеты продолжают поступать, они начинают уничтожаться. Данная ситуация позволяет административно ограничивать доступную полосу пропускания.

Накопленные токены позволяют пропускать короткие всплески, но при продолжительном превышении пакеты будут задерживаться, а в крайнем случае — уничтожаться.

8.2.3.3. Алгоритмы управления перегрузками

К алгоритмам управления перегрузками (QoS Congestion Management) относятся FIFO Queueing, PQ, WFQ, CBWFQ, LLQ.

8.2.3.3.1. FIFO Queueing. *Обработка трафика в порядке поступления пакетов (First In First Out, FIFO)* является самым простым подходом к планированию очереди. Потеря пакетов происходит лишь при переполнении буфера. Задержка и потеря пребывающих пакетов зависят от интервала времени между двумя поступлениями соседних пакетов, а также от их длины. Уменьшение интервала между поступлениями пакетов и/или увеличение длины пакета приводит к росту очереди.

При дисциплине обслуживания в порядке поступления в очередь все пакеты обрабатываются без приоритетов. Поэтому различным информационным потокам невозможно предоставить разное качество обслуживания.

8.2.3.3.2. Priority Queue. *Механизм приоритетной обработки трафика (Priority Queue)* (рис. 8.3) предусматривает разделение всего сетевого трафика на небольшое количество классов с назначением каждому классу приоритета. Поступивший в период перегрузки пакет помещается в одну из очередей согласно его приоритету (количество очередей соответствует числу классов).

Приоритеты очередей имеют абсолютный характер предпочтения при обработке: пока из более приоритетной очереди не будут выбраны все пакеты, устройство не переходит к обработке следующей, менее приоритетной.

Конечный размер буферной памяти сетевого устройства предполагает некоторую предельную длину каждой очереди. Пакет, поступивший в то время, когда буфер заполнен, просто отбрасывается.

Приоритетное обслуживание очередей обеспечивает высокое качество сервиса для пакетов из самой приоритетной очереди. Если средняя интенсивность их поступления в устройство не превосходит пропускной способности выходного интерфейса (и производительности внутренних блоков самого устройства, участвующих в продвижении пакетов), то пакеты с наивысшим приоритетом всегда

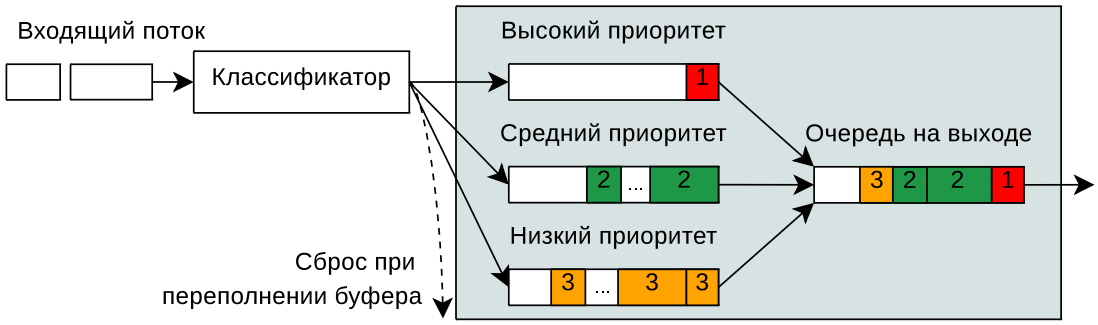


Рис. 8.3. Механизм Priority Queue

получают ту пропускную способность, которая им необходима. Качество обслуживания пакетов остальных классов ниже, чем у пакетов с наивысшим приоритетом.

Приоритетное обслуживание обычно применяется в том случае, когда в сети есть чувствительный к задержкам трафик, но его интенсивность невелика, так что его наличие не слишком ущемляет остальной трафик.

8.2.3.3.3. Stochastic Fairness Queueing. *Алгоритм стохастического справедливого обслуживания (Stochastic Fairness Queueing, SFQ)* поровну распределяет между сеансами доступную полосу пропускания. Трафик делится на достаточное количество очередей типа FIFO, по одной на каждый сеанс. После этого, все очереди обрабатываются в циклическом порядке, тем самым обеспечивая каждому сеансу равные шансы на передачу данных [1].

Следует заметить, что SFQ эффективен только в случае, если исходящий интерфейс полностью загружен. В противном случае очередь будет отсутствовать и, следовательно, никакого положительного эффекта наблюдаться не будет.

8.2.3.3.4. Weighted Queuing *Алгоритм взвешенного обслуживания (Weighted Queuing, WQ)* разработан для того, чтобы для всех классов трафика можно было предоставить определённый минимум пропускной способности или удовлетворить требования к задержкам. Под весом какого-либо класса понимается доля выделяемой данному виду трафика пропускной способности выходного интерфейса.

Как и при приоритетном обслуживании, трафик делится на несколько классов, и для каждого вводится отдельная очередь пакетов. С каждой очередью связывается доля пропускной способности выходного интерфейса, гарантируемая данному классу трафика при перегрузках этого интерфейса (рис. 8.4).

Поставленная цель достигается благодаря тому, что очереди обслуживаются последовательно и циклически, и в каждом цикле из каждой очереди забирается такое число байт, которое соответствует весу очереди. В результате каждому классу трафика достаётся гарантированный минимум пропускной способности, что во многих случаях является более желательным результатом, чем подавление низкоприоритетных классов высокоприоритетным.

[1] SFQ называется «стохастической», так как на самом деле для каждого сеанса очередь не формируется, а трафик делится на ограниченное количество очередей на основе хеш-алгоритма.

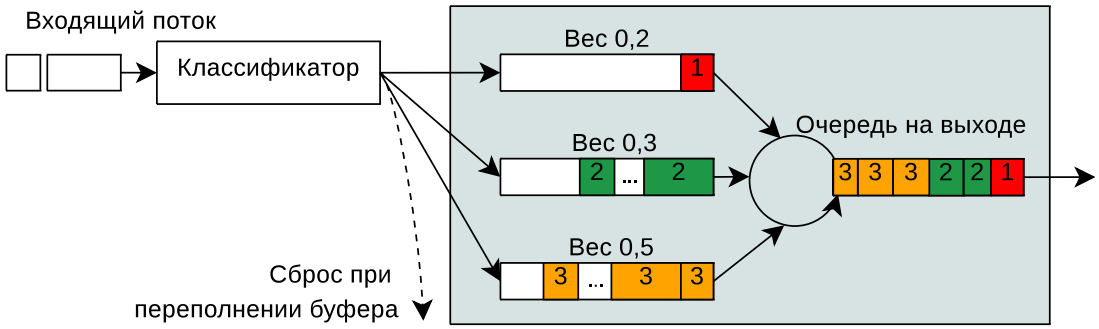


Рис. 8.4. Механизм Weighted Queuing

В общем случае взвешенное обслуживание приводит к большим задержкам и их отклонениям, чем первоочередное обслуживание для самого приоритетного класса, даже при значительном превышении выделенной пропускной способности над интенсивностью входного потока данного класса. Но для более низких приоритетных классов взвешенное справедливое обслуживание часто оказывается более приемлемым с точки зрения создания благоприятных условий обслуживания всех классов трафика.

8.2.3.3.5. Weighted Fair Queuing. *Взвешенное справедливое обслуживание (Weighted Fair Queuing, WFQ)* (рис. 8.5) — это комбинированный механизм обслуживания очередей, сочетающий приоритетное обслуживание со взвешенным.

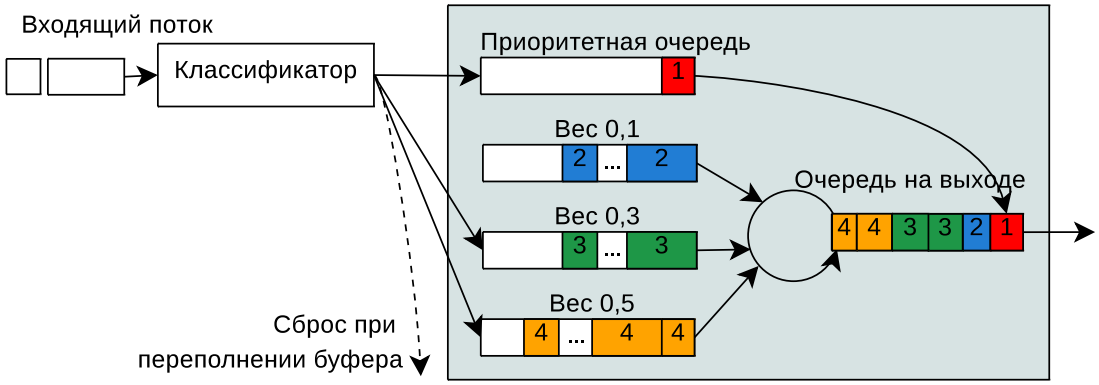


Рис. 8.5. Механизм Weighted Queuing

Производители сетевого оборудования предлагают многочисленные собственные реализации WFQ, отличающиеся способом назначения весов и поддержкой различных режимов работы. Наиболее распространённая схема предусматривает существование одной особой очереди, которая обслуживается по приоритетной схеме — всегда в первую очередь и до тех пор, пока все заявки из неё не уйдут на обслуживание. Эта очередь предназначена для системных сообщений, сообщений управления сетью и, возможно, пакетов наиболее критических и требовательных приложений. Предполагается, что её трафик имеет невысокую интенсивность, поэтому значительная часть пропускной способности выходного интерфейса остаётся другим классам трафика. Остальные очереди устройство про-

сматривает последовательно, в соответствии с алгоритмом взвешенного обслуживания.

Одним из вариантов реализации WFQ является *WFQ, основанный на потоках (Flow-based WFQ, FWFQ)*. В маршрутизаторе создаётся столько очередей, сколько потоков существует в трафике. Под потоком в данном случае понимаются пакеты с определёнными значениями IP-адресов отправителя и получателя и/или портов TCP/UDP отправителя и получателя, а также пакеты с одинаковыми значениями поля ToS. Каждому потоку соответствует отдельная выходная очередь, для которой в периоды перегрузок механизм WFQ выделяет равные доли пропускной способности порта.

Другим вариантом реализации WFQ является *WFQ, основанный на классах (Class-based WFQ, CWFQ или CBWFQ)*. Отличие от обычного WFQ заключается в механизме распределения трафика по классам, который может осуществляться на базе групп QoS, соответствующих набору признаков из списка управления доступом (ACL), или на базе значений поля ToS заголовка пакета.

Во многих сетевых устройствах механизм WFQ является одним из основных для поддержки качества обслуживания, в том числе и в случае различных протоколов, использующих методы сигнализации для координированного поведения всех устройств сети.

8.2.3.3.6. Low Latency Queuing. *Алгоритм обработки очередей с малой задержкой (Low Latency Queuing, LLQ)* [50] является модификацией CBWFQ. Алгоритм (подобно PQ) использует выделенную очередь для обработки трафика, чувствительного к задержке. Остальной трафик обрабатывается по алгоритму CBWFQ.

8.2.3.3.7. Weighted Round Robin. *Дисциплина взвешенного циклического обслуживания (Weighted Round Robin, WRR)* распределяет трафик по классам, используя схему взвешенного циклического обхода. Все классы получают ширину полосы пропускания, пропорциональную присвоенным им весам (рис. 8.6).

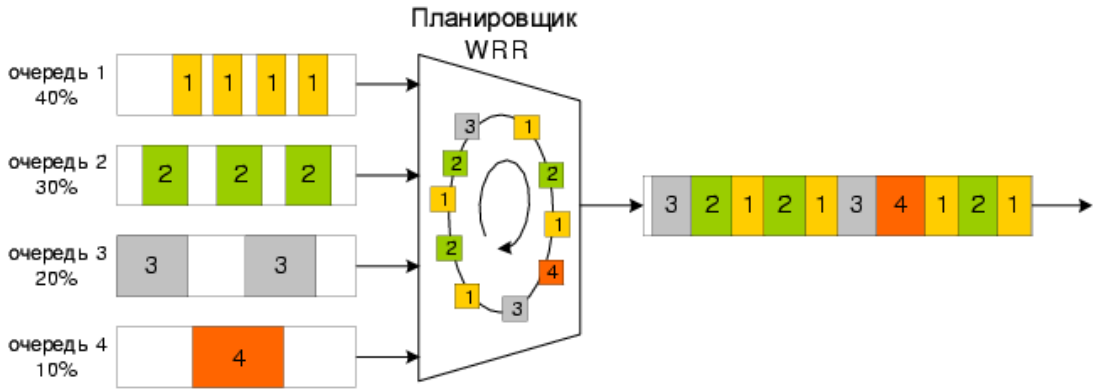


Рис. 8.6. Дисциплина WRR

8.2.3.4. Алгоритмы избежания перегрузок

К алгоритмам избежания перегрузок (QoS Congestion Avoidance) относятся RED, RIO, ARED и др.

8.2.3.4.1. Random Early Detection. Алгоритм *случайного раннего обнаружения (Random Early Detect, RED)* [51] позволяет контролировать нагрузку с помощью выборочного случайного уничтожения некоторых пакетов до полного заполнения очереди.

При поступлении пакета вычисляется значение средней длины очереди $\bar{q}$, на основе которого с учётом двух пороговых значений r_1 и r_2 вычисляется вероятность сброса $\pi(q)$ (рис. 8.7):

$$\pi(\bar{q}) = \begin{cases} 0, & 0 \leqslant \bar{q} < r_1, \\ \dfrac{\bar{q} - r_1}{r_2 - r_1}\pi_{\max}, & r_1 \leqslant \bar{q} \leqslant r_2, \\ 1, & \bar{q} > r_2, \end{cases} \tag{8.1}$$

где $\pi_{\max}$ — параметр, задающий максимальное значение вероятности сброса.

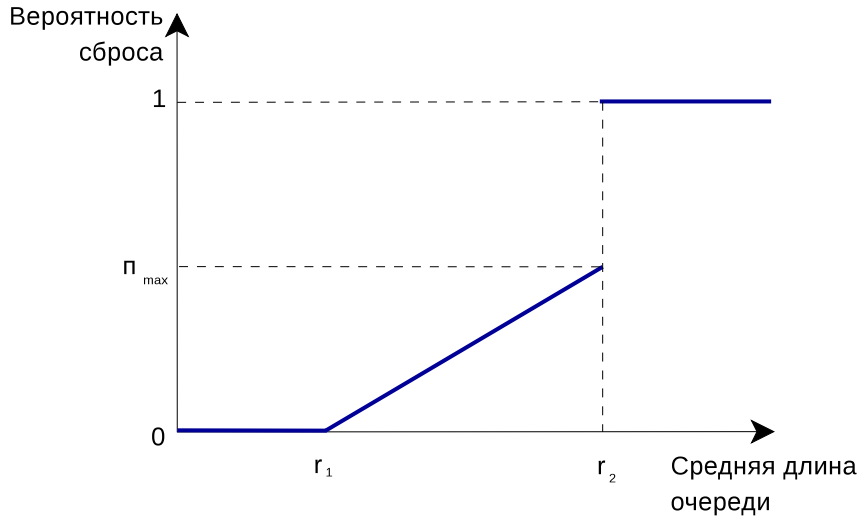


Рис. 8.7. График изменения значений вероятности сброса пакетов в алгоритме RED

При вычислении значения средней длины очереди $\bar{q}$ учитываются текущий размер очереди и предыдущее значение средней длины очереди. Причём, если очередь при поступлении пуста, то

$$\bar{q} = (1 - w_q)\bar{q}_{\text{пред}} + w_q q, \tag{8.2}$$

в противном случае

$$\bar{q} = (1 - w_q)^{f(t - t_q)}\bar{q}_{\text{пред}}, \tag{8.3}$$

где q — текущий размер очереди, w_q — вес очереди, $\bar{q}_{\text{пред}}$ — предыдущее значение средней длины очереди, f — линейная функция времени, t — текущее время, t_q — момент времени, с которого очередь пуста.

8.2.3.4.2. RED with In / Out. В алгоритме *RED with In / Out (RIO) — случайное раннее обнаружение с профильными / непрофильными пакетами* поступающие пакеты делятся на профильные (IN) и непрофильные (OUT). Пакеты поступающего трафика определяются как IN-пакеты, если трафик находится в пределах заданной политики, и как OUT-пакеты, если трафик вышел за пределы заданной политики.

Для принятия решения о сбросе OUT-пакетов используется алгоритм RED относительно средней длины общей очереди, а для принятия решения о сбросе IN-пакетов используется алгоритм RED относительно средней длины виртуальной очереди только из IN-пакетов.

8.2.3.4.2.1. RED with In / Out and Coupled Virtual Queues. Алгоритм *RED with In / Out and Coupled Virtual Queues (RIO-C) — случайное раннее обнаружение с профильными / непрофильными пакетами и парными виртуальными очередями* так же, как и алгоритм RIO, оперирует понятиями IN- и OUT-пакетов. В данном алгоритме для пакетов с различным приоритетом используются разные ограничения, что позволяет пакетам с более высоким приоритетом обслуживаться быстрее. Кроме того, в алгоритме используется задание вероятности сброса пакета, которая может увеличиваться при увеличении интенсивности трафика какого-либо приоритета.

Принятие решения о сбросе IN-пакетов с приоритетом j, $j < n$ зависит от средней длины виртуальной очереди, состоящей только из IN-пакетов с приоритетом, меньшим или равным j.

Принятие решения о сбросе OUT-пакетов с приоритетом n зависит от средней занятости общей (физической) очереди.

8.2.3.4.3. Adaptive RED. Основная идея алгоритма *Adaptive RED (RED)* [52] заключается в адаптации параметра $\pi_{\max}$ так, чтобы значение средней длины очереди находилось между пороговыми значениями r_1 и r_2, но лежало в интервале $[0,01; 0,5]$.

Алгоритм адаптации параметра $\pi_{\max}$ следующий. Для заданного интервала времени, если текущее значение средней длины очереди $\bar{q} > \bar{q}_{\text{target}}$ и $\pi_{\max} \leqslant 0,5$, то $\pi_{\max}$ увеличивается на величину $\alpha = \min(0,01, \pi_{\max}/4)$. В противном случае, если текущее значение средней длины очереди $\bar{q} < \bar{q}_{\text{target}}$ и $\pi_{\max} \geqslant 0,01$, то $\pi_{\max}$ умножается на величину $\beta = 0,9$. При этом $(r_1 + 0,4(r_2 - r_1)) \leqslant \bar{q}_{target} \leqslant (r_1 + 0,6(r_2 - r_1))$.

Таким образом, ARED устраняет зависимость RED от параметра $\pi_{\max}$, поскольку его значение не фиксируется.

8.2.4. Технология IntServ и протокол RSVP

Модель с интеграцией услуг (Integrated Services, IntServ) [53, 54] была разработана для обслуживания единичных потоков, которым предоставляется два вида услуг: *услуга передачи с гарантированной битовой скоростью (Guaranteed Bit Rate Service)* [55] и *услуга передачи с управляемой нагрузкой (Controlled Load Service)* [56].

Услуги с управляемой нагрузкой обеспечивают гарантию того, что зарезервированный поток достигнет своего пункта назначения с минимальным вмешательством со стороны трафика, доставляемого без гарантий (применяются при

передаче трафика Internet-приложений, чувствительных к перегрузкам в сети, например к FTP).

Услуга гарантированной битовой скорости обеспечивает ограничение задержки при передаче без отбрасывания дейтаграмм, удовлетворяющих параметрам трафика, в условиях отсутствия сбоев в работе сетевых компонентов или изменений в информации о маршрутах во время жизни потока. Эта услуга гарантирует минимальное вмешательство со стороны трафика, доставляемого без гарантий, изоляцию зарезервированных потоков и числовое выражение максимальной задержки. Услуга применяется для тех приложений масштаба реального времени, которые позволяют воспроизводить аудио- и видеофайлы.

8.2.4.1. Компоненты модели IntServ

Основные компоненты модели IntServ (рис. 8.8):

— *классификатор трафика (Packet Classifier)* — распределяет поступающие пакеты по классам обслуживания;
— *модуль управления доступом (Flow Admission Control)* — принимает решения о возможности получения трафиком требуемого количества ресурсов, не влияя при этом на ранее предоставленные гарантии;
— *диспетчер пакетов (Packet Scheduler)* — обеспечивает обработку пакетов в соответствии с приоритетом и дисциплиной обслуживания очередей (например, при помощи алгоритмов Drop Tail, RED, WFQ и т.п.);
— *модуль резервирования ресурсов (Flow Resource Reservation)* — обеспечивает управление другими модулями, по запросу выполняет необходимое резервирование ресурсов и поддерживает его вплоть до момента окончания выполнения процедуры резервирования.

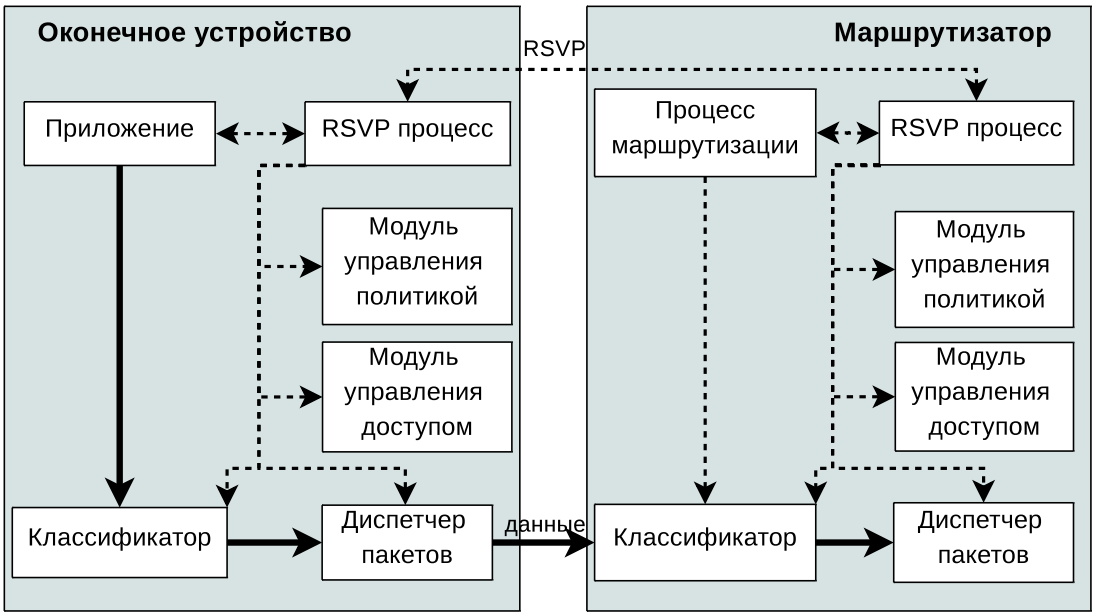


Рис. 8.8. Основные компоненты IntServ

За работу модуля резервирования ресурсов в модели IntServ отвечает *протокол резервирования ресурсов (Resource Reservation Protocol, RSVP)* [57, 58]. Данный протокол позволяет зарезервировать определённую долю сетевых ресурсов,

необходимую информационному потоку, на протяжении всего маршрута от станции отправителя до станции получателя. Кроме того, протокол RSVP содержит описание фильтра и идентификатора резервируемого потока, необходимые при распределении потоков трафика по классам.

Резервирование ресурсов для потока делится на *индивидуальное (Distinct Reservations)* и *общее (Shared Reservations)*.

Индивидуальное резервирование применяется в тех приложениях, в которых сразу несколько источников могут одновременно генерировать потоки данных. В этом случае каждый поток нуждается в отдельном управлении доступом и планировании очереди на всем пути к получателю. Для таких потоков необходимо осуществлять отдельное резервирование ресурсов для каждого отправителя и каждого канала в пути.

Общее резервирование применяется в тех приложениях, в которых несколько источников данных передают информацию неодновременно (например, цифровые аудиоприложения). Такой поток не нуждается в отдельном резервировании ресурсов для каждого отправителя, для него достаточно одного резервирования, которое при необходимости можно будет применить к любому отправителю в группе.

8.2.4.2. Формат сообщений RSVP

Все сообщения RSVP начинаются с общего заголовка, за которым следует тело сообщения, состоящее из переменного числа объектов переменной длины.

Рис. 8.9. Формат общего заголовка сообщений RSVP

Поле *Версия (Version)* (длина 4 бита) содержит номер версии протокола.

Поле *Флаги (Flags)* (длина 4 бита) зарезервировано, и флаги пока не определены.

Поле *Тип сообщения (Message Type)* (длина 8 бит) указывает на один из типов передаваемых сообщений: 1 = Path, 2 = Resv, 3 = PathErr, 4=ResvErr, 5=PathTear, 6 = ResvTear, 7 = ResvConf.

Поле *Контрольная сумма RSVP (RSVP checksum)* (длина 16 бит) содержит контрольную сумму сообщения.

Поле *Send_TTL* (длина 8 бит) указывает на время жизни (TTL) протокола IP, с которым было послано сообщение.

Поле *Длина RSVP (RSVP length)* (длина 16 бит) содержит значение полной длины RSVP сообщения в байтах, включая общий заголовок, и объекты переменной длины, которые за ним следуют.

В RSVP определены следующие типы сообщений:

— *сообщение определения пути (Path)* содержит определение формата пакетов данных, спецификацию характеристик трафика потока, информацию о потоке, IP-адрес источника, адрес места назначения для текущей сессии, адрес предшествующего узла;

— *сообщение резервирования (Resv)* содержит запросы резервирования от узла к узлу, от получателей к отправителям в направлении, противоположном движению потока данных;
— *сообщение отмены прохода (PathTear)* аннулирует состояние прохода и модифицирует состояние резервирования в узле;
— *сообщение отмены Resv (ResvTear)* удаляет соответствующие состояния резервирования;
— *сообщение об ошибке прохода (PathErr)* содержит данные об ошибке в обрабатываемых сообщениях Path;
— *сообщение об ошибках резервирования (ResvErr)* сообщает об ошибках при обработке сообщений Resv или о спонтанном нарушении резервирования, например, в результате административного вмешательства;
— *сообщение подтверждения (ResvConf)* посылается в ответ на запрос подтверждения резервирования.

8.2.4.3. Компоненты RSVP. Механизм резервирования ресурсов

Основными компонентами RSVP являются: отправитель, получатель, маршрутизаторы и хосты, находящиеся на пути от получателя к отправителю, потоки (совокупность IP-пакетов, посылаемых отправителем одному или более получателям, с соответствующим потоку идентификатором — FlowLabel).

Рассмотрим механизм работы протокола RSVP.

Перед началом посылки потока IP-пакетов, требующего определённого качества обслуживания, отправитель при помощи сообщения Path информирует получателя о желании начать передачу и о необходимых параметрах качества (поле FlowSpec).

В ответ получатель рассылает заявки на резервирование ресурсов всем узлам, находящимся на выбранном маршруте. Данная заявка содержит в себе, помимо IP-адреса отправителя и получателя, поля FlowSpec (параметры качества), AdmissionControl (информация о возможности отправителя предоставить требуемое качество) и PolicyControl (характеризует права получателя на проведение операции резервирования QoS). Если поля AdmissionControl и PolicyControl установлены верно, то узел, получающий данную заявку, резервирует требуемые ресурсы.

Если все узлы смогли зарезервировать ресурсы и заявка на резервирование дошла до отправителя, он начинает передачу потока данных.

В случае невозможности зарезервировать запрашиваемые ресурсы отправитель получает соответствующее уведомление. Если передачи потока не происходит, узел с зарезервированным качеством обслуживания ждёт фиксированное время, после чего освобождает ресурсы.

Механизм RSVP выглядит следующим образом:

— отправитель посылает сообщение Path;
— получатель в ответ посылает заявку на резервирование QoS на маршрутизаторы между от правителем и получателем;
— каждый маршрутизатор, получив заявку, проверяет поля AdmissionControl и PolicyControl и в случае их достоверности резервирует требуемый QoS;
— отправитель после получения уведомления о резервировании QoS начинает передачу потока пакетов.

8.2.4.4. Функциональность RSVP

Перечислим функциональные возможности протокола RSVP:

— RSVP выполняет резервирование для уникастных и мультикастных приложений, динамически адаптируясь к изменениям членства в группе вдоль маршрута;
— RSVP является симплексным протоколом, т.е. он выполняет резервирование для однонаправленного потока данных;
— RSVP ориентирован на получателя, т.е. получатель данных инициирует и поддерживает резервирование ресурсов для потока;
— RSVP поддерживает динамическое членство в группе и автоматически адаптируется к изменениям маршрутов;
— RSVP не является маршрутным протоколом, но зависит от существующих и будущих маршрутных протоколов;
— RSVP транспортирует и поддерживает параметры управления трафиком и политикой, которые остаются непрозрачными для RSVP;
— RSVP обеспечивает несколько моделей резервирования, для того чтобы удовлетворить требованиям различных приложений;
— RSVP обеспечивает прозрачность операций для маршрутизаторов, которые его не поддерживают;
— RSVP может работать с IPv4 и IPv6.

8.2.4.5. Преимущества и недостатки модели IntServ

Преимущества модели IntServ:

— протокол RSVP, на котором базируется IntServ, не зависит от протоколов маршрутизации, т.к. сам рассылает вдоль зарезервированного пути управляющие сообщения и контролирует таким образом состояние маршрута;
— IntServ может обеспечить гарантированную величину задержки, если трафик имеет более или менее определённую природу (например, соответствует каким-то ограничениям) и маршрутизаторы поддерживают дисциплину обслуживания WFQ;
— IntServ при помощи WFQ может обеспечить контролируемое совместное использование каналов (перегружающий канал трафик ограничен выделенной ему шириной полосы пропускания, но если у канал имеет остаточную ёмкость, то по нему разрешается передавать любую комбинацию трафика).

Недостатки модели IntServ:

— плохая масштабируемость RSVP, особенно в высокоскоростных магистральных сетях, поскольку объём ресурсов, которые необходимы маршрутизатору для обработки и хранения информации RSVP, увеличивается пропорционально количеству потоков QoS, вследствие чего возрастает нагрузка на маршрутизаторы;
— для поддержания резервирования ресурсов протокол RSVP вынужден постоянно рассылать обновляющие сообщения для сохранения состояния вдоль зарезервированных путей.

8.2.5. DiffServ

Модель дифференцированных услуг (Differentiated Services, DiffServ или DS) была предложена IETF в 1997–1998 гг. [59, 60, 61, 62]. Модель DiffServ предлагает

простой, но довольно эффективный метод приоритезации трафика в соответствии с требованиями различных приложений.

При разработке модели DiffServ преследовались следующие цели:

— универсальность — широкое разнообразие комплексных услуг должно быть чётким и понятным, а сетевые услуги должны быть независимы от приложений;
— простота — целая система или её часть не должны зависеть от передачи сигналов для индивидуальных потоков;
— эффективность — данные об индивидуальных потоках или клиентах не должны быть использованы в промежуточных узлах сети.

8.2.5.1. Архитектура DiffServ

Модель DiffServ реализуется путём создания подсетей, называемых *DiffServ-доменами*.

DiffServ-домен (или DS-домен) — множество смежных DS-узлов, работающих в соответствии с определёнными наборами политик обслуживания трафика и согласованными множествами правил пошагового обслуживания групп PHB в каждом узле.

Per-Hop-Behavior (PHB) — сценарий пошаговой обработки трафика на узле коммутации.

В DiffServ-домен могут входить:

— обычные станции (инициализируют и/или потребляют трафик);
— граничные маршрутизаторы:
  – соединяют DiffServ-домены друг с другом или конечных пользователей с DiffServ-доменом;
  – классифицируют трафик;
  – задают политику обработки каждого класса;
  – обрабатывают трафик в соответствии с заданной политикой;
— центральные (внутренние) маршрутизаторы:
  – соединяют граничные маршрутизаторы с внутренними или внутренние маршрутизаторы между собой;
  – осуществляют транзит трафика через DiffServ-домен с поддержкой дифференцирования услуг.

8.2.5.2. DiffServ Code Point

Для приоритезации трафика в соответствии с требованиями различных приложений модель DiffServ использует *кодовое слово (DiffServ Code Point, DSCP)*, необходимое для выбора *правил пошаговой обработки (Per-Hop Behavior, PHB)*, которой пакет подвергается в каждом узле.

DSCP располагается в поле *Тип обслуживания (Type of Service — ToS)* заголовка IPv4 или в поле *Класс трафика (Class of Traffic — CoT)* заголовка IPv6 (рис. 8.10; рис. 5.2) [59].

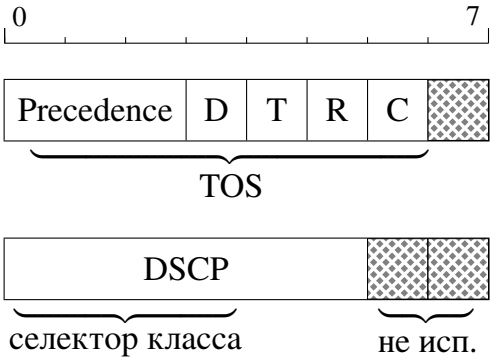


Рис. 8.10. Поле TOS заголовка IPv4 и DSCP

Для кодировки DSCP используются шесть младших бит, а два старших зарезервированы для дальнейшего развития технологии, и их значения должны игнорироваться DS-системами[1]. Исторически три младших бита поля типа IP-сервиса применялись для обозначения относительного приоритета данного пакета. В целях сохранения частичной обратной совместимости с более ранними системами в архитектуре DS предусмотрено резервирование восьми *типов локального поведения (Per Hop Behavior — PHB)* для обслуживания пакетов с DSCP вида «xxx000», где «x» принимает значения 0 или 1. Такие кодовые слова называют *селекторами класса*. Так, PHB, соответствующие кодовому слову DSCP=«11x000», обслуживают пакеты с большим приоритетом, чем PHB, соответствующие DSCP=«000000»[2].

8.2.5.3. Per Hop Behavior

Поддержка определённых типов PHB или их групп реализуется путём применения различных алгоритмов управления очередями и буферным пространством.

Двумя основными типами локального поведения являются *Expedited Forwarding (EF)* [62] и *Assured Forwarding (AF)* [61].

При использовании PHB EF поток передаётся с минимальным колебанием задержки, значением задержки и потерями, гарантируется определённая пропускная способность.

При использовании PHB AF поток передаётся с количественно определёнными параметрами качества обслуживания (гарантия высокой вероятности доставки данных). Важным требованием к сервису AF является сохранение первоначального порядка следования пакетов, принадлежащих к одному микропотоку.

Определено четыре класса PHB AF. Для каждого класса узел DiffServ выделяет ограниченное количество ресурсов: определённую пропускную способность канала и объём буферного пространства. В пределах каждого класса AF пакеты маркируются тремя возможными уровнями вероятности потерь. В случае возникновения перегрузки в первую очередь отбрасываются пакеты, отмеченные тем

[1] Последние 2 бита применяются для реализации механизма *Явное предупреждение о перегрузке (Explicit Congestion Notification — ECN)*.

[2] Кодовое слово «000000» рекомендовано для обозначения трафика типа Best Effort.

кодом DSCP, который соответствует более высокой вероятности потерь. При возникновении перегрузки для сервисов класса AF могут использоваться ресурсы, отведённые под другие классы, если отнесённый к ним трафик отсутствует.

Значения DSCP, рекомендованные для маркировки пакетов группы AF, приведены в табл. 8.1.

Таблица 8.1

Группа типов локального поведения AF

Вероятность потери	Класс 1		Класс 2		Класс 3		Класс 4	
Низкая	AF11	001010	AF21	010010	AF31	011010	AF41	100010
Средняя	AF12	001100	AF22	010100	AF32	011100	AF42	100100
Высокая	AF13	001110	AF23	010110	AF33	011110	AF43	100110

8.2.5.4. Основные функциональные блоки DiffServ

DS-узел для реализации описанных выше функций использует определённые функциональные блоки. Диаграмма взаимосвязи функциональных блоков узла сети DiffServ (рис. 8.11) может быть представлена устройством с одним входом и одним или более выходами, обладающими своим набором параметров контроля. Стоит отметить, что приведённая на рис. 8.11 модель не обязательно содержит все пять элементов. Обычно данная модель соответствует модели, действующей в пределах DS-домена, но так же может соответствовать и модели конкретного DS-узла.

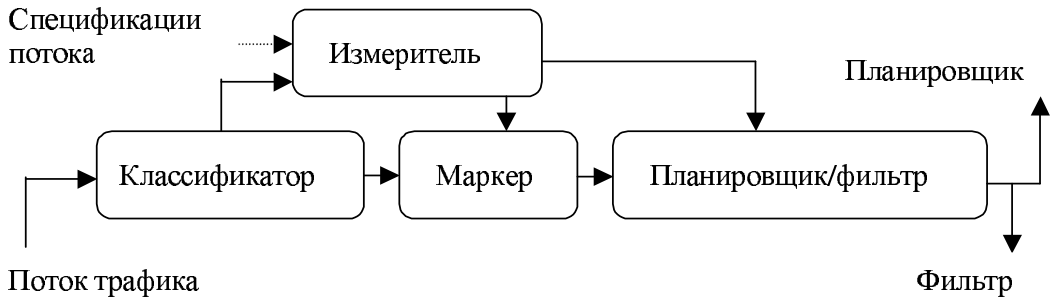


Рис. 8.11. Основные функциональные блоки DiffServ

Классификатор (Classifier) задаёт правила, определяющие подмножество трафика, которое может получить дифференцированное обслуживание за счёт кондиционирования и/или отображения на один или несколько агрегированных потоков внутри домена DiffServ.

Кондиционирование (упорядочивание) трафика необходимо для того, чтобы гарантировать соответствие трафика, попадающего в домен DiffServ, правилам,

определённым в *соглашении об упорядочивании трафика (Traffic Conditioning Agreement, TCA)* согласно политике выделения ресурсов домена.

Кондиционирование трафика включает следующие операции:

- *измерение (Metering)* — измерение временны́х параметров (например, интенсивности) трафика, выбранного из потока классификатором;
- *формирование (Shaping)* — процесс задержки пакетов в потоке для того, чтобы согласовать их с некоторым определённым профилем передачи трафика;
- *профилирование и/или перемаркировка (Policing / Re-mark)* — процесс сбрасывания или задержки / перемаркировки пакетов в потоке данных в соответствии с показателями измерителя и параметрами профиля трафика.

Входящая в состав спецификации сервиса спецификация кондиционирования трафика содержит:

- детализацию параметров сервиса (скорость, значение задержки, вероятность потери пакетов);
- описание топологических границ предоставления сервиса;
- спецификации трафика (ограничения на потребляемые ресурсы в виде параметров алгоритма Token Bucket);
- описание способа обработки трафика, значения характеристик которого превышают заявленные в спецификации;
- описание услуг по маркировке трафика;
- описание услуг по формированию трафика.

Контроллер определяет, соответствуют ли параметры трафика его профилю. Результаты проверки для конкретного пакета могут использоваться для инициирования операций *маркировки*, *отбрасывания* или *формирования*.

Маркировщик (Marker) присваивает каждому пакету конкретное кодовое значение (DSCP), включая таким образом маркированный пакет в конкретный агрегированный поток.

В качестве примера маркировщика можно привести алгоритм *скользящего временного окна с 2-цветным маркером — Time Sliding Window with Two Color Marking (TSW2CM)*.

Формирователь (Shaper) обеспечивает соответствие потока его профилю посредством задержки обслуживания пакетов при помощи формирования очереди. Формирователь обычно имеет буфер ограниченного размера, так что пакеты могут быть отброшены из-за нехватки в буфере места для размещения задержанных пакетов.

В качестве примера формирователя можно привести дисциплины *Token Bucket Filter (TBF)*, *Stochastic Fairness Queueing (SFQ)*, *Class Based Queueing (CBQ)*, *Hierarchical Token Bucket (HTB)*, *Weighted Round Robin (WRR)*.

Отбраковщик (Dropper) удаляет некоторые или все пакеты в потоке, чтобы обеспечить его соответствие профилю. Этот процесс называется *приведением потока в соответствие с требованиями политики* или *профилированием*.

В качестве примера отбраковщиков можно привести алгоритмы семейства *Random Early Detection (RED)*.

При выходе пакетов из модуля кондиционирования трафика пограничного узла DiffServ поле DSCP каждого пакета должно иметь соответствующее значение.

8.3. QoS в ATM

Спецификации и рекомендации по управлению трафиком и качеством обслуживания в сетях ATM были разработаны организациями ATM Forum и ITU.

Качество обслуживания в сетях ATM обеспечивается путём явного определения конечными станциями соглашения, описывающего характеристики их трафика.

Дескриптор потока содержит следующие характеристики трафика:

— *значение пиковой скорости передачи ячеек (Peak Cell Rate, PCR)*;
— *значение поддерживаемой скорости передачи ячеек (Sustainable Cell Rate, SCR)*;
— *значение минимальной скорости передачи (Minimum Cell Reta, MCR)*;
— *значение максимального размера всплеска (Maximum Burst Size, MBS)*;
— *допустимое отклонение времени задержки передачи ячейки (Cell Delay Variation Tolerance, CDVT)*.

Конечные станции формируют трафик путём буферизации данных и их передачи в рамках соглашения по QoS. ATM-коммутаторы, в свою очередь, проверяют все характеристики проходящего через них трафика и сравнивают их с соглашением по QoS. При отклонении от соглашения по QoS коммутатор устанавливает CLP-бит для всего несогласующегося с заданными параметрами трафика, что увеличивает вероятность его сброса в случае перегрузки.

В ATM определены две стратегии сброса:

— *частичное отбрасывание пакетов (Partial Packet Diskard, PPD)*;
— *раннее отбрасывание пакетов (Early Packet Diskard, EPD)*.

В случае, если сбрасывается ATM-ячейка, являющаяся частью большого пакета, функция PPD отбрасывает и остальные ячейки пакета, поскольку пересборка сегментированных пакетов невозможна и необходимо переслать заново весь пакет целиком. Функция PPD производит частичное отбрасывание ячеек пакета, поскольку их отбрасывание может произойти после того, как предыдущие ячейки этого пакета были поставлены в очередь на отправку.

Функция EPD применяется до того, как ячейка встаёт в очередь на отправку. При поступлении пакета EPD проверяет коэффициент загрузки выходного буфера. Если он ниже сконфигурированного порогового значения, то все ячейки пакета помещаются в очередь. В противном случае ATM-коммутатор считает, что возможно переполнение буфера и пакет не сможет быть помещён в буфер полностью. Поэтому происходит отбрасывание всего пакета.

Для предотвращения перегрузок в сетях ATM используются следующие *механизмы управления трафиком*:

— *управление установлением соединения (Connection Admission Control, CAC)*;
— *управление параметрами трафика и QoS (Usage Parammeter Control, UPC)*;
— *формирование трафика (Traffic Shaping)*;
— *обобщённый алгоритм контроля скорости передачи ячеек (Generic Cell Rate Algorithm, GCRA)*.

Спецификация ATM Forume определяет для ATM четыре категории сервиса для разных типов трафика:

— *постоянная скорость передачи (Constant Bit Rate, CBR)*;
— *переменная скорость передачи (Variable Bit Rate, VBR)*:
  – *VBR реального времени (Real-Time VBR, RT-VBR)*;
  – *VBR без требований реального времени (Non-Real-Time VBR, NRT-VBR)*;

— *доступная скорость передачи (Available Bit Rate, ABR)*;
— *неопределённая скорость передачи (Unspecified Bit Rate, UBR)*.

Категория CBR применяется для чувствительного к задержкам трафика (аудио и видео). Передача данных осуществляется с постоянной скоростью и малыми задержками, но требует резервирования части полосы пропускания.

Категория RT-VBR используется, если требуется жёсткая синхронизация между ячейками и поддержка чувствительного к задержкам трафика. Категория NRT-VBR применяется для допускающего задержки трафика, без синхронизации между ячейками. Категории VBR не резервируют полосу пропускания, но и не могут гарантировать качества сервиса.

Категория ABR применяется для передачи трафика, допускающего задержки (передача данных), даёт возможность многократно использовать виртуальные каналы, обеспечивает для соединения допустимые значения ширины полосы пропускания и коэффициента потерь.

Категория UBR применяется для трафика, допускающего задержки и потери пакетов, не резервирует полосу пропускания для виртуального канала, может использовать один виртуальный канал для нескольких передач. UBR не гарантирует качества сервиса.

Аналогично рекомендация ITU-T 1.371 определяет для ATM следующие *категории сервиса* для разных типов трафика:

— *детерминированная скорость передачи (Deterministic Bit Rate, DBR)*,
— *статистическая передача (Statistic Bit Rate, SBR)*,
— *доступная скорость передачи (Available Bit Rate, ABR)*,
— *немедленный перенос блока ATM (ATM block Transfer with Immediate Transmission, ABT/IT)*,
— *перенос блока ATM с задержкой (ATM block Transfer with Delayed Transmission, ABT/DT)*.

Рекомендация I.363 ITU-T определяет следующие, соответствующие категориям сервиса, *адаптационные уровни (ATM Adaptation Level, AAL)*:

— AAL1 и AAL2 соответствуют услугам, требующим постоянную или переменную скорость передачи;
— AAL3 и AAL4 соответствуют услугам с ориентацией и без ориентации на соединение;
— AAL5 соответствует услуге передачи данных, объединённых в пакеты.

Уровень адаптации ATM реализует адаптацию функций уровня к требованиям передачи информационных потоков, исходящих от различных приложений. Для каждого соединения ATM задаются следующие *основные параметры QoS*:

— *коэффициент потерь ячеек (Cell Loss Ratio)*;
— *задержка передачи ячейки (Cell Transfer Delay)*;
— *вариации задержек при передаче ячеек (Cell Delay Variation, CDV)*[1];

дополнительные параметры QoS:

— *отношение числа ячеек с ошибками к общему числу переданных ячеек (Cell Error Ratio, CER)*;
— *доля ячеек, принимаемых не по адресу назначения (Cell Misinsertion Rate)*;
— *коэффициент ошибочных блоков (Severely-Errored Cell Block Ratio)*.

[1]Большая величина CDV приводит к прерыванию аудио- и видеосигналов.

8.4. Организация виртуальных каналов при помощи меток (MPLS)

Для обеспечения QoS в сетях на базе MPLS используется комбинация двух технологий — DiffServ и MPLS Traffic Engineering.

Значение DSCP, необходимое для маркировки пакетов, размещается в трёхбитовом поле Experimental заголовка MPLS.

Traffic Engineering (TE) — представляет собой механизм управления направлением прохождения трафика с целью выполнения определённых условий (резервирование каналов, распределение загрузки сети, балансировка и предотвращение перегрузок).

Для задания пути прохождения определённого типа трафика механизм TE в MPLS использует *однонаправленные туннели (MPLS TE Tunnel)*. Технологически MPLS TE основывается на формировании маршрутов прохождения пакетов (LSP) через сеть с помощью механизма создания *туннелей (MPLS Tunnel)*, который в свою очередь базируется на *стекировании меток (Labels Stack)*.

Рассмотрим этапы MPLS TE:

1) организация MPLS домена — определяется сетевая топология, состоящая из набора маршрутизаторов и каналов с определёнными свойствами между ними (полоса пропускания и прочее);
2) наложение ограничений — указываются минимальные требования к сети, такие как начальные и конечные точки прохождения трафика, графы путей между ними и методы вычисления явных и динамических маршрутов по ним, требуемая полоса пропускания;
3) изучение параметров сетевой среды — формируется база связей (линков) между всеми маршрутизаторами и их состояниями (Link State Database);
4) вычисление путей прохождения трафика — на граничных входных (по отношению к потоку трафика) маршрутизатора выполняется специальный алгоритм *Constrained Base Algorithm*, учитывающий политику выбора лучшего пути для LSP туннеля (возможности каналов, границы MPLS домена, полоса пропускания);
5) установление путей — при помощи специального протокола сигнализации (RSVP-ext или CR-LDP) устанавливаются просчитанные пути;
6) установление маршрутов с учётом туннелей TE при помощи IGP — LSP туннели начинают работать как интерфейсы, указывая путь (туннель) прохождения трафика через MPLS-домен;
7) продвижение пакетов — с помощью механизма Label Stacking происходит обеспечение необходимого туннелирования и продвижение пакетов.

Глава 9. Мультисервисные сети

Мультисервисная сеть представляет собой инфраструктуру, использующую единый канал для передачи данных разных типов трафика.

Далее перечислены аспекты построения мультисервисных сетей:

- *конвергенция загрузки сети* — передача различных типов трафика в рамках единого формата представления данных;
- *конвергенция протоколов* — переход от множества существующих сетевых протоколов к общему;
- *физическая конвергенция* — передача различных типов трафика в рамках единой сетевой инфраструктуры;
- *конвергенция устройств* — построение архитектуры сетевых устройств, способной в рамках единой системы поддерживать разнотипный трафик;
- *конвергенция приложений* — интеграция различных функций в рамках единого программного средства;
- *конвергенция технологий* — создание единой общей технологической базы для построения сетей связи, способной удовлетворить требованиям и региональных сетей связи, и локальных вычислительных сетей;
- *организационная конвергенция* — централизация сетевых, телекоммуникационных, информационных служб под управлением менеджеров высшего звена.

Конвергирование подразумевает объединение двух направлений — коммутацию каналов (передачу голоса) и коммутацию пакетов (передачу данных). Поэтому далее сначала рассматривается мультисервисная сеть на базе коммутации каналов (ISDN), затем технологии для организации видеоконференций и других услуг по различным типам сетей передачи данных (H.323), сеть сигнализации № 7 как сеть управления мультисервисными сетями. Затем рассматриваются две концепции (Softswitch и IMS), имеющие возможность предоставления медиауслуг поверх различных сетей передачи. В заключение определяется понятие сетей следующего поколения (NGN), рассматриваются архитектура и концепции построения.

9.1. Цифровая сеть с интеграцией служб (ISDN)

Цифровая сеть с интеграцией служб (Integrated Services Data Network, ISDN) представляет собой сеть с коммутацией каналов (телефонную сеть), обеспечивающую полностью цифровые соединения между оконечными устройствами для поддержания широкого спектра информационных услуг.

ISDN обеспечивает единый интерфейс доступа к цифровой сети передачи данных для устройств, выполняющих широкий набор задач, с сохранением полной прозрачности сети для пользователей. Основное назначение ISDN — передача 64-Кбит/с по 4 КГц проводной линии и обеспечение интегрированных телекоммуникационных услуг.

9.1.1. Каналы ISDN

ISDN поддерживает три типа логических цифровых коммуникационных каналов, которые выполняют следующие функции:

B-канал используется для передачи информации (данные, видео и голос);
D-канал используется для передачи сигнализации и пакетов данных между пользовательским оборудованием и сетью;
H-канал выполняет те же самые функции, что и D-канал, однако работает при скорости, превышающей DS-0 (64 Кбит/с).

9.1.2. Устройства ISDN

В число компонентов ISDN входят:
— *терминальный адаптер (Terminal Adapter, TA)* — используется для подключения не-ISDN устройств к сети ISDN;
— *локальная станция (Local Exchange, LE)* — используется в телефонной станции, работает с протоколом ISDN и является частью сети;
— *локальное окончание (Local Termination, LT)* — используется для обозначения LE, служащих для работы с Local Loop (абонентским шлейфом);
— *оконечная станция (Exchange Termination, ET)* — используется для обозначения LE, отвечающих за функции коммутации;
— *сетевое оконечное оборудование (Network Termination, NT)*:
  – NT1 — служит для завершения соединений между пользователем и LE, отвечает за работу, мониторинг, подачу питания и мультиплексирование каналов;
  – NT2 — любое устройство, применяемое пользователем для коммутации, мультиплексирования и концентрации (локальная сеть, компьютер, терминальный контроллер и т. д.);
— *терминальное оборудование (Terminal Equipment, TE)* — любое пользовательское устройство (например, телефон или факсимильный аппарат).

9.1.3. Структура кадров ISDN

Кадры ISDN имеют следующую структуру (рис. 9.1).

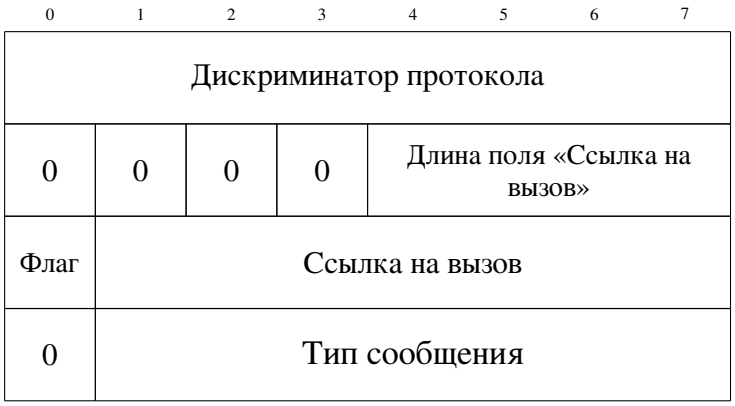


Рис. 9.1. Формат кадра ISDN

Поле *Дискриминатор протокола* указывает протокол, используемый для оставшейся части.

Поле *Длина поля «Ссылка на вызов»* определяет длину следующего поля, которое может занимать один или два октета (в зависимости от типа используемого кодирования).

Поле *Флаг* имеет нулевое значение для сообщений, передаваемых стороной, выделяющей значения ссылки на вызов, 1 — в остальных случаях.

Поле *Ссылка на вызов* используется устройствами для идентификации соединения между устройством, инициировавшим вызов, и коммутатором ISDN.

Поле *Тип сообщения* определяет тип передаваемого сообщения и может занимать один или два (для специфических сообщений) октета. В двухоктетных сообщениях первый октет содержит восемь нулей.

9.1.4. Услуги сетей ISDN

Услуги идентификации номера:
- предоставление идентификации вызывающей линии;
- ограничение идентификации вызывающей линии;
- предоставление идентификации подключённой линии;
- ограничение идентификации подключённой линии;
- предоставление идентификации вызываемой линии;
- ограничение идентификации вызываемой линии;
- определение злонамеренного вызова;
- идентификация вызывающего при ожидании вызова.

Услуги, связанные с адресацией:
- прямой набор для УАТС;
- немедленный вызов к фиксированному адресату;
- задержанный вызов к фиксированному адресату;
- подадресация;
- преселекция;
- взаимодействующие номера;
- сокращённый адрес;
- несколько номеров у линии.

Услуги по завершению вызова:
- возврат вызова;
- ожидание вызова;
- завершение вызова по неответу;
- завершение вызова к занятому абоненту;
- перехват;
- услуга постановки в очередь.

Услуги переадресации:
- переадресация вызова безусловная;
- переадресация вызова при занятости;
- переадресация вызова по неответу;
- переадресация вызова на записанное сообщение;
- избирательная переадресация;
- передача вызова к звонку / к занято;
- отклонение вызова;
- ограниченная переадресация вызова.

Тарификация дополнительных услуг:
- информация о тарифе при установлении соединения;
- информация о тарифе в ходе разговора;

— информация о тарифе по завершении вызова;
— немедленный расчёт;
— бесплатный номер.

Услуги по ограничению:
— замкнутая группа пользователей;
— запрет входящего вызова;
— запрет исходящего вызова;
— не беспокоить;
— выборочный отказ от вызова;
— ограниченная передача вызова;
— приоритет;
— катастрофический приоритет;
— аварийная ситуация;
— редакция списка просмотра;
— выборочный приём вызова.

Услуги для нескольких участников:
— конференция с добавлением;
— конференция «Встреть меня»;
— трёхсторонняя конференция;
— удержание вызова;
— поиск линии.

Услуги мобильности:
— переносимость терминала;
— дистанционное управление;
— групповой поиск.

Специальные услуги предупреждений:
— вызов по тревоге;
— приоритетный вызов к охране.

Услуги, связанные с УАТС:
— прямой набор;
— общий и индивидуальный набор.

Управление дополнительными услугами:
— управление абонентами;
— удалённый доступ;
— общая пассивизация.

Дополнительные услуги:
— сигнализация от пользователя к пользователю;
— услуга передачи данных;
— подслушивание / перехват.

9.2. Сеть на базе стека H.323

Серия рекомендаций H.32x предназначена для организации видеоконференций по различным типам сетей передачи данных (табл. 9.1).

H.323 — это рекомендации ITU-T [63] для мультимедийных приложений в вычислительных сетях, не обеспечивающих гарантированное качество обслуживания (QoS). Такие сети включают в себя сети пакетной коммутации IP и IPX на базе Ethernet, Fast Ethernet и пр. Рекомендация H.323 регламентирует технические требования к коммутации речи, видео и данных по пакетным сетям, а также к связи с сетями с коммутацией каналов.

Таблица 9.1

Сводная таблица протоколов семейства H.32x

Рекомендация	**H.320**	**H.321**	**H.322**	**H.323**	**H.324**
Год принятия	1990	1995	1995	1996	1996
Сеть	Узкополосная ISDN	Широкополосная ISDN, ATM	Сеть с коммутацией пакетов и гарантированным QoS (isoEthernet)	Сеть с коммутацией пакетов и негарантированным QoS (Ethernet)	Аналоговые телефонные сети (PSTN или POTS)
Видео	H.261 H.263	H.261 H.263	H.261 H.263	H.261 H.263 H.264/AVC	H.261 H.263
Аудио	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.728	G.711 G.722 G.723.1 G.726 G.728 G.729	G.723
Мультиплексирование	H.221	H.221	H.221	H.225.0	H.223
Управление	H.230 H.242	H.242	H.242 H.230	H.245	H.245
Поддержка многоточечных конференций	H.231 H.243	H.231 H.243	H.231 H.243	H.323	-
Обмен данными	T.120	T.120	T.120	T.120	T.120
Сетевой интерфейс	I.400	AAL I.363 AJM I.361 PHY I.400	I.400 & TCP/IP	TCP/IP	V.34 модем

9.2.1. Архитектура сети H.323

Архитектура сети H.323 представлена на рис. 9.2.

Объектами сети H.323 являются:

— *терминал (Terminal)* — оконечное мультимедийное устройство, обеспечивающее возможность двусторонней коммуникации речи, видео или данных с другим объектом сети в реальном времени;

— *межсетевой шлюз (Gateway)* — устройство, предназначенное для преобразования мультимедийной и управляющей информации при сопряжении разнородных сетей;

— *устройство управления многоточечными соединениями (Multipoint Control Unit, MCU)* — предназначено для организации конференций с участием трёх и более участников;

— *контроллер зоны (Gatekeeper)* — рекомендуемое, но не обязательное устройство, обеспечивающее сетевое управление и исполняющее роль виртуальной телефонной станции;

— *разграничитель (Border Element)* — элемент сети H.323, посредством которого выполняется коммуникация между административными доменами.

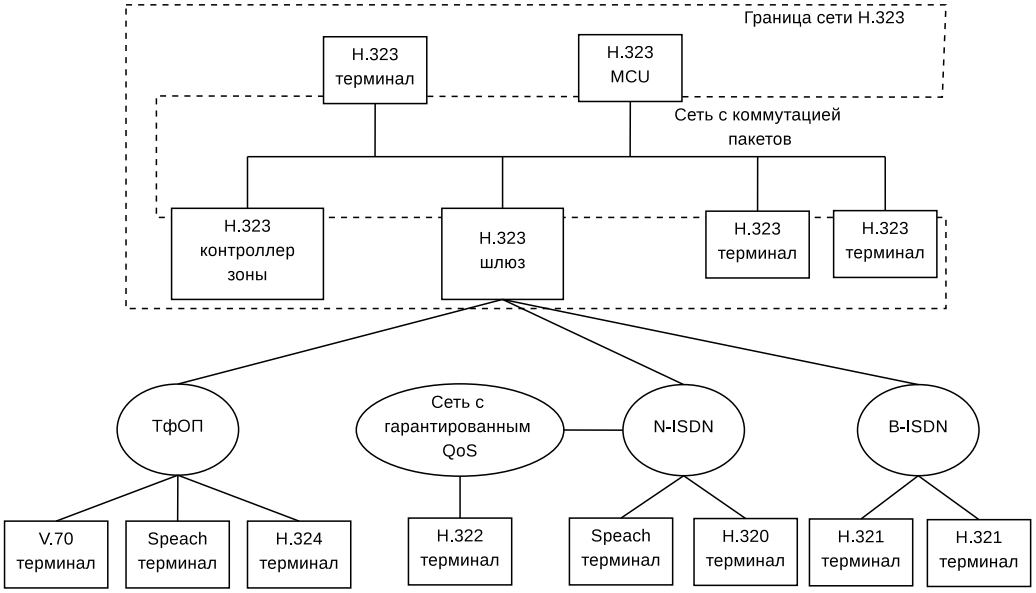


Рис. 9.2. Архитектура сети H.323

Терминал H.323 обеспечивает звуковую связь и может дополнительно поддерживать передачу видео или данных. Терминал H.323 может быть реализован как программное приложение на персональном компьютере или как самостоятельное устройство (например, телефон).

Терминал должен поддерживать следующие протоколы:

— H.245 для согласования параметров соединения;
— Q.931 для установления соединения и согласования параметров этого соединения;
— RAS (Registration/Admission/Status) для взаимодействия с контроллером зоны;
— RTP/RTCP для работы с потоками аудио и видеопакетов;
— семейство протоколов H.450;
— аудиокодек G.711 для сжатия аудиопотока.

Дополнительно терминал может поддерживать другие аудиокодеки, а также видеокодеки H.261 и/или H.263. Необязательной является поддержка протокола совместной работы над документами T.120.

Межсетевой шлюз не является обязательным компонентом сети H.323 и используется только в том случае, когда требуется установить соединение с терминалом, расположенным в сети другого типа (стандарта). Связь обеспечивается трансляцией протоколов установки и разрыва соединений, а также форматов передачи данных. Основной функцией межсетевого шлюза H.323 является преобразование сигнализационных протоколов, способа передачи, процедур коммуникации и способа кодирования, что обеспечивает возможность взаимодействия пользователей разных технологий. Шлюзы H.323 широко применяются в IP-телефонии для сопряжения IP-сетей и цифровых или аналоговых коммутируемых телефонных сетей (ISDN (Integrated Services Digital Network) или PSTN (Public Switched Telephone Network)).

Межсетевой шлюз может выполнять следующие функции:

— функция PSTN-терминала — содержит PSTN сигнализационный интерфейс, которым заканчивается PSTN сигнализация, и PSTN медиаинтерфейс, которым заканчивается медиапоток;
— функция H.323-терминала — содержит VoIP сигнализационный интерфейс, которым заканчивается H.323 сигнализация (H.225, H.245), и интерфейс пакетной передачи, которым заканчивается медиапоток, передаваемый пакетами протокола RTP;
— преобразование сигнализационных протоколов, используемых в H.323 и PSTN-сетях;
— преобразование медиапотоков, сформированных при помощи различных алгоритмов сжатия;
— управление связью — координирование сигнализационных потоков и преобразование медиапотоков, в том числе и установление, изменение и разъединение соединения между медиапотоками в PSTN и IP-сети в течение вызова.

Контроллер зоны также не является обязательным компонентом сети H.323, и если используется, то обеспечивает сетевое управление и выполняет функции виртуальной телефонной станции. В этом случае контроллер зоны становится центральной точкой для всех обращений внутри одной зоны — совокупности терминалов, шлюзов и серверов MCU, управляемых одним контроллером.

Контроллер зоны выполняет следующие функции:

— основные:
  – трансляция адресов — преобразование внутренних адресов сети и телефонных номеров формата E.164 (применяются в сетях ISDN) в транспортные адреса протоколов IP или IPX;
  – управление доступом — авторизация доступа в H.323-сеть путём обмена RAS-сообщениями «запрос регистрации» (ARQ), «удовлетворение запроса» (ACF) и «отклонение запроса» (ARJ);
  – управление полосой пропускания — используются RAS-сообщения «запрос ширины полосы пропускания» (BRQ), «удовлетворение запроса» (BCF) и «отклонение запроса» (BRJ);
  – управление зоной H.323 — установление вызова, использование ресурсов разрешается исключительно тем объектам сети H.323, которые зарегистрированы как члены зоны определённого контроллера зоны;
— дополнительные:
  – управление процессом установления соединений — обработка служебных сообщений протокола сигнализации Q.931 [64];
  – авторизация соединения;
  – управление вызовами — контроль за состоянием всех активных соединений, что позволяет обеспечить выделение необходимой полосы пропускания и баланс загрузки сетевых ресурсов за счёт переадресации вызовов на другие терминалы и шлюзы;
  – тарификация — хранение и обработка информации о вызовах и предоставленных услугах.

Устройство MCU предназначено для поддержки конференции между тремя и более участниками. В этом устройстве должен присутствовать *контроллер Multipoint Controller (MC)* и, возможно, *процессоры Multipoint Processors (MP)*. Контроллер MC поддерживает протокол H.245 [65] и предназначен для согласования параметров обработки аудио- и видеопотоков между терминалами. Процессоры занимаются коммутированием, микшированием и обработкой этих потоков.

Стандарт H.323 определяет три типа конференцсвязи между тремя или более числом терминалов и межсетевых шлюзов: *централизованная*, *децентрализованная*, *гибридная*.

Централизованная многоточечная конференция требует наличия устройства MCU. Каждый терминал обменивается с MCU потоками аудио, видео, данными и командами управления по схеме «точка–точка». Контроллер MC, используя протокол H.245, определяет возможности каждого терминала. Процессор MP формирует необходимые для каждого терминала мультимедийные потоки и рассылает их. Кроме того, процессор может обеспечивать преобразования потоков от различных кодеков с различными скоростями данных.

Децентрализованная многоточечная конференция использует технологию групповой адресации. Участвующие в конференции H.323 терминалы осуществляют многоадресную передачу мультимедиа потока остальным участникам без посылки на MCU. Передача контрольной и управляющей информации осуществляется по схеме «точка–точка» между терминалами и MCU. В этом случае контроль многоточечной рассылки осуществляется контроллером MC.

Гибридная схема организации конференцсвязи является комбинацией двух предыдущих. Участвующие в конференции H.323 терминалы осуществляют многоадресную передачу только аудио- или только видеопотока остальным участникам без посылки на MCU. Передача остальных потоков осуществляется по схеме «точка–точка» между терминалами и MCU. В этом случае задействуются как контроллер, так и процессор MCU.

9.2.2. Адресация элементов сети H.323

Адресация терминалов VoIP в основном основывается на буквенно-цифровых последовательностях, распознаваемость которых обеспечена иерархической организацией группы серверов. Однако из-за потребности интеграции услуг между сетями PSTN и VoIP каждому абоненту PSTN должна быть обеспечена возможность адресации VoIP абонента, и наоборот.

Стандарт H.323 поддерживает следующие типы адресов:

- dialedDigits (в старых версиях E.164) — цифровой идентификатор в виде телефонного номера;
- h323-ID — имя пользователя или адрес электронной почты (e-mail address);
- url-ID — общий тип адреса (включает H.323-URL и PSTN-URL);
- transport-ID — транспортный адрес оконечного оборудования;
- email-ID — адрес электронной почты;
- partyNumber — цифровой идентификатор;
- mobile-UIM — идентификатор мобильных пользователей с возможностью взаимодействия с мобильными сетями общего пользования 2G и 3G.

Такой подход требует отдельного преобразования и распознавания адреса, а также особых процедур регистрации, обеспечиваемых контроллерами зоны H.323 и разграничителями.

В случае адресации с помощью цифр телефонного номера используются префикс зоны и технологический префикс, однозначно определяющие зону административного домена.

Каждое устройство в сети H.323 может иметь более одного адреса (возможно, одного и того же типа). Единственное условие — все адреса одного устройства должны ссылаться на уникальный транспортный адрес этого устройства.

9.2.3. Основные характеристики H.323

Основные характеристики H.323:

- независимость от сети — возможна работа поверх существующих архитектур сетей;
- управление шириной полосы — каждому H.323-вызову выделяется определённая ширина полосы;
- независимость от приложения и платформы — не требуется применения определённой аппаратной или программной платформы;
- поддержка многосторонних конференций;
- взаимодействие — прозрачная коммутация для конечного пользователя;
- гибкость — в одной H.323 конференции могут участвовать терминалы различных возможностей коммуникации.

9.2.4. Обработка звуковых сигналов (Audio Signal)

Одним из важных факторов эффективного использования пропускной способности IP-канала является выбор оптимального алгоритма кодирования / декодирования речевой информации — кодека.

Типы речевых кодеков по принципу действия можно разделить на три группы:

1) кодеки с импульсно-кодовой модуляцией (ИКМ) и адаптивной дифференциальной импульсно-кодовой модуляцией (АДИКМ):
   - разработаны в конце 1950-х годов,
   - используются в системах традиционной телефонии;
2) кодеки с вокодерным[1] преобразованием речевого сигнала:
   - разработаны для снижения требований к пропускной способности радиотракта в системах мобильной связи,
   - применяется гармонический синтез сигнала на основании информации о его вокальных составляющих — фонемах,
   - реализованы как аналоговые устройства;
3) комбинированные (гибридные) кодеки:
   - сочетают в себе технологию вокодерного преобразования / синтеза речи, но оперируют с цифровым сигналом посредством специализированных преобразователей цифровых сигналов,
   - содержат в себе ИКМ или АДИКМ кодек и реализованный цифровым способом вокодер.

[1] Вокодер — электронный цифровой музыкальный инструмент, преобразующий звук человеческого голоса путём изменения его волновых и частотных характеристик.

9.2.4.1. Оценка MOS

MOS (Mean Opinion Score) — средняя экспертная оценка разборчивости речи — метод субъективного тестирования качества речи, часто используемый для сравнения характеристик речевых кодеков, при котором слушатели выставляют оценки по пятибалльной системе. Результирующая оценка MOS вычисляется как среднее арифметическое для большого числа оценок.

Таблица 9.2

Оценки MOS

Качество	Оценка MOS
высокое	$4,0$–$5,0$
стандартное телефонное	$3,5$–$4,0$
приемлемое	$3,0$–$3,5$
синтезированный звук	$2,5$–$3,0$

9.2.4.2. G.711

Рекомендация G.711 [66] описывает кодек, использующий преобразование аналогового сигнала с точностью 8 бит, тактовой частотой 8 КГц и простейшей компрессией амплитуды сигнала. Скорость потока данных на выходе преобразователя составляет 64 Кбит/с (8 бит x 8 КГц). Для снижения шума квантования и улучшения преобразования сигналов с небольшой амплитудой при кодировании используется нелинейное квантование по уровню согласно специальному псевдологарифмическому закону. Существуют два основных алгоритма, представленных в стандарте:

1) μ-law (используется в Северной Америке и Японии):
прямое преобразование:

$$F(x) = \operatorname{sgn}(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 \leqslant x \leqslant 1;$$

обратное преобразование:

$$F^{-1}(y) = \operatorname{sgn}(y)\frac{1}{\mu}\left[(1+\mu)^{|y|}-1\right], \quad -1 \leqslant y \leqslant 1,$$

где $\mu = 255$ (8 бит).
2) A-law (используется в Европе и в остальном мире):
прямое преобразование:

$$F(x) = \operatorname{sgn}(x)\begin{cases} A\dfrac{|x|}{1+\ln(A)}, & |x| \leqslant \dfrac{1}{A}, \\ \dfrac{1+\ln(A|x|)}{1+\ln(A)}, & \dfrac{1}{A} \leqslant |x| \leqslant 1; \end{cases}$$

обратное преобразование:

$$F^{-1}(y) = \operatorname{sgn}(y) \begin{cases} \dfrac{|y|(1+\ln(A))}{A}, & |y| \leqslant \dfrac{1}{1+\ln(A)}, \\ \dfrac{\exp(|y|(1+\ln(A))-1)}{A}, & \dfrac{1}{1+\ln(A)} \leqslant |y| < 1, \end{cases}$$

где $A = 87,6$ — параметр сжатия.

Оба алгоритма являются логарифмическими, но более поздний A-law был изначально предназначен для компьютерной обработки процессов.

Типичная оценка MOS составляет $4,2$. Обычно любое устройство VoIP поддерживает этот тип кодирования.

Кодек G.711 широко распространён в системах традиционной телефонии с коммутацией каналов. Несмотря на то, что рекомендация G.711 в стандарте H.323 является основной и первичной, в шлюзах IP-телефонии данный кодек применяется редко из-за высоких требований к полосе пропускания и задержкам в канале передачи. Использование G.711 в системах IP-телефонии обоснованно лишь в тех случаях, когда требуется обеспечить максимальное качество кодирования речевой информации при небольшом числе одновременных разговоров.

9.2.4.3. G.723.1

Кодек G.723.1 [67] является одним из базовых кодеков сжатия речи, утверждённым ITU-T в рекомендации G.723.1 в ноябре 1995 г. Кодек предназначен для приложений IP-телефонии, в частности, для организации видеоконференций по телефонным сетям. Рекомендация G.723.1 является частью более общего стандарта H.324, описывающего подход к организации видеоконференций, при этом целью является обеспечение видеоконференций с использованием обычных модемов.

Кодек G.723.1 представляет собой комбинацию аналого-цифрового преобразования / цифро-аналогового преобразования и вокодера. Применение вокодера позволяет снизить скорость передачи данных в канале, что принципиально важно для эффективного использования как радиотракта, так и IP-канала.

Кодек G.723.1 осуществляет преобразование аналогового сигнала в поток данных со скоростью 64 Кбит/с (ИКМ), а затем при помощи многополосного цифрового фильтра / вокодера выделяет частотные фонемы, анализирует их и передаёт по IP-каналу информацию только о текущем состоянии фонем в речевом сигнале. Данный алгоритм преобразования позволяет снизить скорость кодированной информации до $5,3$–$6,3$ Кбит/с без видимого ухудшения качества речи.

Кодек G.723.1 предусматривает два режима работы: $6,3$ Кбит/с (кадр имеет размер 189 бит, дополненных до 24 байт) и $5,3$ Кбит/с (кадр имеет размер 158 бит, дополненных до 20 байт). Первый режим применяется для сетей с пакетной передачей голоса и использует алгоритм сжатия речи MP-MLQ (Multipulse Maximum Likelihood Quantization — многоимпульсное квантование с максимальным правдоподобием), который позволяет добиться весьма существенного сжатия речевой информации при сохранении достаточно высокого качества звучания. Второй режим применяется в сетях со смешанным типом трафика (голос / данные) и использует алгоритм CELP (Code Excited Linear Prediction — кодирование с линейным предсказанием). Режим работы кодека G.723.1 может меняться динамически от кадра к кадру.

Алгоритм CELP [68, 69] построен на модели кодирования с использованием процедуры «анализа через синтез», линейного предсказания и векторного квантования. CELP-анализ состоит из трёх основных процедур:

— кратковременное линейное предсказание;
— долговременный поиск по адаптивной кодовой книге;
— поиск по стохастической кодовой книге.

CELP-синтез состоит из этих же процедур, выполненных в обратном порядке.

Кодер оперирует с кадрами речевого сигнала длиной 30 мс, дискретизованными с частотой 8 КГц. Для каждого кадра производится анализ речевого сигнала и выделяются передаваемые параметры CELP-модели: 10 линейных спектральных пар (несут информацию о коэффициентах фильтра линейного предсказания), индексы и коэффициенты усиления в адаптивной и фиксированной кодовых книгах. Далее эти параметры кодируются в битовый поток и передаются в канал.

В декодере эта битовая посылка используется для восстановления параметров сигнала возбуждения и коэффициентов синтезирующего фильтра. Далее восстанавливается речь путём пропускания сигнала возбуждения через синтезирующий фильтр. Затем для улучшения качества восприятия синтетического сигнала выходной сигнал с фильтра-синтезатора пропускается через постфильтр.

Длительность кадров кодека G.723.1 составляет 30 мс с длительностью предварительного анализа сигнала $7,5$ мс.

Оценка MOS для данного кодека составляет $3,9$ в режиме $6,3$ Кбит/с и $3,7$ — в режиме $5,3$ Кбит/с.

9.2.4.4. Кодек G.726

Рекомендация G.726 основана на алгоритме кодирования ADPCM — адаптивная дифференциальная ИКМ. Этот алгоритм даёт практически такое же качество воспроизведения речи, как и ИКМ, однако для передачи информации при его использовании требуется полоса всего 16–32 Кбит/с. Кодек предназначен для использования в системах видеоконференций; в приложениях IP-телефонии этот кодек практически не применяется. Оценка по MOS составляет $4,3$.

9.2.4.5. G.728

Алгоритм G.728 стандартизован ITU в 1992 г. [70], основан на методе LD-CELP (Low-Delay Code Excited Linear Prediction — кодирование с линейным предсказанием и низкой задержкой) и предназначен для сжатия и передачи речевых данных со скоростью 16 Кбит/с, при этом внося задержку при кодировании от 3 до 5 мс.

Алгоритм применяется к цифровой последовательности, получаемой в результате аналого-цифрового преобразования речевого сигнала с 16-разрядным разрешением. Входной сигнал с частотой дискретизации 8кГц, сжатый по A- или μ-закону (см. раздел 9.2.4.2), преобразуется для получения линейного кода.

Оценка MOS для данного кодека составляет $3,6$.

Предназначен для использования в основном в системах видеоконференций. В устройствах IP-телефонии данный кодек применяется достаточно редко.

9.2.4.6. G.729

В основе кодеков G.729 [71, 72, 73] лежит алгоритм CS-ACELP (Conjugate Structure – Algebraic Code Excited Linear Prediction) — сопряжённая структура с управляемым алгебраическим кодированием с линейным предсказанием. Процесс преобразования вносит задержку 15 мс. Скорость кодированного речевого сигнала составляет 8 Кбит/с.

Алгоритм основан на модели кодирования с использованием линейного предсказания по алгебраической кодовой книге (CELP-модель). Кодер оперирует с кадрами речевого сигнала длительностью 10 мс, дискретизованными с частотой 8КГц, что соответствует 80 16-битным отсчётам в линейном законе. Для каждого кадра производится анализ речевого сигнала и выделяются параметры модели (коэффициенты фильтра линейного предсказания, индексы и коэффициенты усиления в адаптивной и фиксированной кодовых книгах). Далее эти параметры кодируются и передаются в канал.

В декодере битовая посылка используется для восстановления параметров сигнала возбуждения и коэффициентов синтезирующего фильтра. Речь восстанавливается путём пропускания сигнала через кратковременный синтезирующий фильтр.

Синтезирующий фильтр имеет полюсную передаточную функцию 10-го порядка. Для работы синтезатора основного тона используется адаптивная кодовая книга. В последующем речь улучшается адаптивной постфильтрацией.

В случае потери передаваемой кодером битовой посылки исходные данные для речевого синтезатора получаются интерполяцией данных с предыдущих не повреждённых кадров, но при этом энергия интерполированного речевого сигнала постепенно уменьшается, что не создаёт особого дискомфорта у слушателя.

В устройствах VoIP и VoFR данный кодек занимает лидирующее положение, обеспечивая наилучшее качество кодирования речевой информации при достаточно высокой компрессии.

9.2.5. Обработка видеосигналов (Video Signal)

Стандарт H.323 [63] устанавливает два формата изображения — CIF (352×288 пиксела) для яркостного сигнала и QCIF (176×144 пиксела), т.е. с $1/4$ частью разрешения CIF, причём частота смены кадров не должна опускаться ниже 24 кадров в секунду.

CIF (Common Itermediate Format — общий формат обмена) представляет собой стандарт видеоизображения с размером кадра 352×288 пиксела и частотой кадров $7,5$, 10, 15 или 30 к/с. Цвет кодируется в формате YUV (представление цвета, при котором каждый элемент изображения представляется тремя компонентами: яркостной и двумя цветоразностными) с разрядностью 8 бит. Производные форматы: QCIF — 176×144 пикселов, subQCIF — 128×96 пикселов, 4CIF — 704×576 пикселов, 16CIF — 1408×1152 пикселов.

Для компрессии/декомпрессии видеосигнала используются кодеки H.261, H.263, H.264. Различаются они способом обработки изображения.

9.2.5.1. H.261

Стандарт H.261 [74] определяет видеокодек H.261 для аудиовизуальных услуг со скоростью $P \times 64$ Кбит/с, где P может меняться в диапазоне от 1 до 30. В данном кодеке реализована комбинация алгоритмов *DCT (Discrete Cosine Transform)* и *Motion Prediction*.

Алгоритм *DCT (Discrete Cosine Transform — дискретное косинус-преобразование, ДКП)* разработан в 1981 г. В [75, 76] даётся следующее определение.

Определение. Пусть дано изображение размером $N \times N$. Тогда прямое ДКП записывается в виде:

$$t(u,v) = c(u)c(v)\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} I(k,l)\cos\frac{(2k+1)u\pi}{2N}\cos\frac{(2l+1)v\pi}{2N},$$

$$c(m) = \begin{cases}\sqrt{\frac{1}{N}}, m = 0,\\ \sqrt{\frac{2}{N}}, m \neq 0,\end{cases} \qquad u,v = \overline{1, N-1},$$

а обратное:

$$I(k,l) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} t(u,v)c(u)c(v)\cos\frac{(2k+1)u\pi}{2N}\cos\frac{(2l+1)v\pi}{2N},$$

$$c(m) = \begin{cases}\sqrt{\frac{1}{N}}, & m = 0,\\ \sqrt{\frac{2}{N}}, & m \neq 0.\end{cases} \qquad k,l = \overline{1, N-1},$$

Здесь коэффициенты $t(u,v)$ — амплитуды пространственных частот изображения.

Дискретное преобразование обладает следующими свойствами:

— некоррелированность коэффициентов — коэффициенты независимы друг от друга, т.е. точность представления одного коэффициента не зависит от любого другого;
— «уплотнение» энергии — преобразование сохраняет основную информацию в малом количестве коэффициентов.

Motion Prediction — предсказание перемещения — техника межкадрового кодирования, применяемая в кодеках для сжатия сигнала движущегося изображения. В последовательности кадров каждый пиксель в текущем кадре перемещён по сравнению с предшествующим кадром. При этом соседние пиксели перемещаются практически одинаково. Кадр делится на блоки пикселей (16×16 или 8×8), и для описания движения пикселей всего блока вычисляется вектор оценки перемещения (*Motion Estimation*). Предсказание перемещения текущего блока, полученное из предшествующего кадра с помощью вектора компенсации перемещения (*Motion Compensation*), сравнивается с настоящим текущим блоком и формируется, если надо, ошибка предсказания (т.е. компенсация неточности предсказания). Для таких блоков передаётся только вектор оценки перемещения и ошибка предсказания, что значительно экономней простой передачи содержимого блока.

9.2.5.2. H.263

Стандарт H.263 [77] разработан в 1995 г. и определяет видеокодек H.263, предназначенный для передачи видеоизображения с малой скоростью (ниже 64 Кбит/с, например, для связи с помощью модема и аналоговых телефонных линий). Кодек H.263 использует технологию H.261 с дополнительными усовершенствованиями, главным образом в области предсказания перемещения. В отличие от

H.261, для которого предсказываемые направления должны лежать в пределах изображения, для H.263 они могут выходить за границы изображения. Это особенно важно при низких скоростях передачи, не являющихся обязательными для стандарта H.261. Кроме того, кодек H.263 позволяет загружать канал связи практически только изменениями картинки.

Дальнейшим развитием проекта являются кодеки H.263v2 (также известный как H.263+ или H.263 1998) и H.263v3 (известный как H.263++ или H.263 2000).

9.2.5.3. H.264

Стандарт H.264 [78] разработан совместно ITU-T и MPEG и является развитием H.263. Он определяет одноимённый кодек H.264, также известный как *AVC (Advanced Video Coding)* и *MPEG-4* [79], который имеет существенно расширенные возможности по сравнению с H.263, вследствие чего стал основным при разработке программного обеспечения для видеоконференций.

Основные характеристики H.264:

— Многокадровое предсказание перемещения кадров:

  – Более гибкое использование сжатых ранее кадров в качестве опорных. Разрешается использование до 32 ссылок на другие кадры, что поднимает эффективность кодирования, так как позволяет кодеру выбирать для компенсации движения между большим количеством изображений.

  – Независимость порядка воспроизведения изображений и порядка опорных изображений, что позволяет кодеру выбирать порядок изображений для компенсации движения и для воспроизведения с высокой степенью гибкости, которая ограничена только объёмом памяти, гарантирующим возможность декодирования. Устранение ограничения также позволяет в ряде случаев устранить дополнительную задержку, ранее связанную с двунаправленным предсказанием.

  – Независимость методов обработки изображений и возможности их использования для предсказания движения, что обеспечивает кодеру большую гибкость и возможность использовать для предсказания движения изображение, более близкое по содержанию к кодируемому.

  – Компенсация движения с переменным размером блока (от 16×16 до 4×4 пикселя) позволяет крайне точно выделять области движения.

  – Вектора движения, выводящие за границы изображения (по аналогии с H.263).

  – Шеститочечная фильтрация компонента яркости для полупиксельного предсказания с целью уменьшения зубчатости краёв и обеспечения большей чёткости изображения.

  – Точность до четверти пиксела при компенсации движения обеспечивает очень высокую точность описания движущихся областей (что особенно актуально для медленного движения).

  – Взвешенное предсказание, позволяющее использовать масштабирование и сдвиг после компенсации движения на величины, указанные кодером. Такая методика может чрезвычайно сильно поднять эффективность кодирования для сцен с изменением освещённости, например, при эффектах затемнения, постепенного появления изображения.

— Пространственное предсказание от краёв соседних блоков для I-кадров (от англ. Intra Pictures). Новая методика экстраполяции краёв ранее декодированных частей текущего изображения повышает качество сигнала, используемого для предсказания.
— Сжатие макроблоков без потерь:
  – Метод представления макроблоков без потерь в ИКМ, при котором видеоданные представлены непосредственно, что позволяет точно описывать определённые области и допускать строгое ограничение на количество закодированных данных для каждого макроблока.
  – Улучшенный метод представления макроблоков без потерь, позволяющий точно описывать определённые области, затрачивая при этом существенно меньше битов, чем ИКМ.
— Гибкие функции чересстрочного сжатия:
  – Адаптивное к изображению кодирование полей (PAFF), позволяющее кодировать каждый кадр как кадр или как пару полей (полукадров) — в зависимости от отсутствия/наличия движения.
  – Адаптивное к макроблокам кодирование полей (MBAFF), позволяющее независимо кодировать каждую вертикальную пару макроблоков (блок 16×32) как прогрессивные или чересстрочные. Позволяет использовать макроблоки 16×16 в режиме разбиения на поля.
— Новые функции преобразования:
  – Точное целочисленное преобразование пространственных блоков 4×4, позволяющее точно разместить разностные сигналы с минимумом шума.
  – Точное целочисленное преобразование пространственных блоков 8×8, обеспечивающее большую эффективность сжатия схожих областей, чем 4×4.
  – Адаптивный выбор кодеком между размерами блока 4×4 и 8×8.
— Дополнительное преобразование Адамара (разложение обрабатываемых сигналов по системе прямоугольных базисных функций), применяемое к дискретно-косинусным коэффициентам основного пространственного преобразования (к коэффициентам яркости и, в особом случае, цветности) для достижения большей степени сжатия в однородных областях.
— Квантование:
  – Логарифмическое управление длиной шага для упрощения распределения битрейта (битовая скорость передачи данных) кодером и упрощённого вычисления обратной функции квантования.
  – Частотно-оптимизированные матрицы масштабирования квантования, выбираемые кодером для оптимизации квантования на основе человеческих особенностей восприятия.
— Внутренний фильтр деблокинга (удаление блочности) в цикле кодирования, устраняющий артефакты (искажение) блочности, часто возникающие при использовании основанных на DCT-техниках сжатия изображений.

— Энтропийное кодирование[1] квантованных коэффициентов трансформации:
  – *Context-adaptive binary arithmetic coding (CABAC) — контекстнозависимое адаптивное бинарное арифметическое кодирование* — алгоритм сжатия без потерь синтаксических элементов видеопотока на основе вероятности их появления.
  – *Context-adaptive variable-length coding (CAVLC) — контекстнозависимое адаптивное кодирование с переменной длиной кодового слова* — альтернатива CABAC меньшей сложности.
  – Часто используемое, простое и высокоструктурированное кодирование словами переменной длины многих элементов синтаксиса, не закодированных CABAC или CAVLC, известное как Exp-Golomb (экспоненциальное кодирование Голомба).
— Функции устойчивости к ошибкам:
  – Определение уровня сетевой абстракции, позволяющее использовать один и тот же синтаксис видео в различных сетевых окружениях, включая наборы параметров последовательности и наборы параметров изображения, которые обеспечивают большую надёжность и гибкость, чем предыдущие технологии.
  – Гибкое упорядочивание макроблоков, также известное как группы частей и произвольное упорядочивание частей — методы реструктурирования порядка представления макроблоков в изображениях.
— Благодаря произвольному упорядочиванию частей новый стандарт позволяет посылать и получать их в произвольном порядке друг относительно друга. Это может снизить задержку в приложениях реального времени.
  – Разбиение данных — функция, обеспечивающая разделение данных разной важности по разным пакетам данных с разными уровнями защиты от ошибок.
  – Избыточные части. Возможность посылки кодером избыточного представления областей изображений, позволяя воспроизвести области изображений, данные о которых были потеряны в процессе передачи.
  – Нумерация кадров, позволяющая создать «подпоследовательности» (включая временное масштабирование включением дополнительных кадров между другими), а также обнаружить (и скрыть) потери целых кадров при сбоях канала или пропаже пакетов.

9.2.6. Конференц-связь для передачи данных (Data)

Стандарт T.120 [80] представляет собой совокупность телекоммуникационных и прикладных протоколов для организации и проведения многоточечной конференции в реальном времени [81].

Данный стандарт регламентирует порядок организации и поддержания конференций на любой платформе, управление множеством участников и программ,

[1] *Энтропийное кодирование* — кодирование словами (кодами) переменной длины, при которой длина кода символа имеет обратную зависимость от вероятности появления символа в передаваемом сообщении.

безошибочный и безопасный обмен данными при различных возможных сетевых сценариях.

В семейство T.120 входят следующие протоколы:

— T.121 представляет основу для разработки прикладных протоколов;
— T.122 совместно с T.125 определяет доступные многоточечные услуги;
— T.123 специфицирует транспортные профили ТфОП, ISDN, цифровых сетей с коммутацией каналов CSDN, цифровых сетей с коммутацией пакетов PSDN, сети Novell NetWare IPX и сети TCP/IP; обеспечивает вышележащим уровням независимость от типа сети и предоставляет четыре канала разного приоритета между двумя точками, что необходимо для обеспечения преимущества пересылки данных реального времени (например, информации о перемещении курсора) перед фоновой передачей данных (например, транспортировкой файлов);
— T.124 регламентирует общий процесс управления конференцией Generic Conference Control (GCC), обеспечивая полный набор инструментов для её организации и управления; в частности, GCC обеспечивает функции ведущего конференции и функции резервирования.
— T.125 описывает многоточечный протокол связи (Multipoint Communication Service Protocol, MCS), задающий процедуры для передачи сигнальной информации и данных между провайдерами MCS; при многоточечном соединении можно ограничить доступ к определённым наборам данных, сделав их доступными лишь для некоторых участников телеконференции;
— T.126 определяет процедуры просмотра и аннотирования неподвижных изображений между двумя или несколькими приложениями;
— T.127 предусматривает средства файлового обмена между участниками конференции, в том числе их одновременную приоритетную передачу, а также опции для сжатия файлов перед их транспортированием;
— T.128 регламентирует аудиовизуальное управление.

Стек протоколов T.120 имеет двухуровневую архитектуру. Протоколы T.122, T.123, T.124 и T.125 образуют нижний уровень и описывают независимый от приложений механизм для организации многоточечной связи. В тоже время, протоколы T.126, T.127 и T.128 располагаются на верхнем уровне и по своей сути являются прикладными протоколами. Следует отметить, что в рамках одной конференции могут сосуществовать как стандартизованные, так и нестандартизованные приложения.

В зависимости от конкретной реализации продукты T.120 могут устанавливать соединения, выполнять передачу и приём данных и работать совместно, используя программное разделение, передачу файлов и др.

9.2.7. Управление (Control)

Совокупная система управления H.323 основывается на трёх отдельных сигнализационных каналах: канале H.245, канале установления вызова и RAS-канале.

Протокол управления мультимедийной конференцией H.245 [65] обеспечивает согласование возможностей компонентов, установление и разрыв логических соединений, передачу запросов на установление приоритета, управление потоком (загрузкой канала), передачу общих команд и индикаторов.

Сообщения протокола H.245 передаются по H.245-каналу управления, используя коммутируемый способ передачи данных с помощью протокола TCP, что гарантирует последовательную передачу данных без ошибок. Между любыми двумя элементами сети можно установить только один H.245-канал.

Межтерминальный обмен параметрами позволяет согласовывать режимы работы и форматы кодирования информации, что обеспечивает взаимодействие терминалов от разных производителей. В процессе обмена сообщениями о параметрах уточняются возможности терминалов по приёму и передаче различных видов трафика.

Все H.245-сигнализационные сообщения принадлежат одной из следующих категорий:

— запрос (Request) — сообщения, которые требуют от получателя выполнения определённых действий, включая и ответ на принятый запрос;
— ответ (Response) — сообщения, которые посылаются в ответ на сообщения из предыдущей категории;
— команда (Command) — команды, которые от получателя требуют выполнения определённых действий, но не включают ответ на команду;
— индикация (Indication) — сообщения информативного типа, которые от получателя не требуют ни действий, ни ответа.

Процедуры H.245:

— объявление о возможности обмена медиа потоками (Capabilities Exchange) — информация, необходимая для выбора поддерживаемого обеими сторонами вида медиа коммуникации;
— определение ведущей стороны в коммуникации (Master Slave Determination) — договорённость о ведущем и ведомых оконечных узлах;
— открытие и закрытие логических каналов сигнализации (Logical Channel Signalling);
— запрос на изменение установленного соединения (Request Mode) — запрос на изменение характеристик медиа потока;
— закрытие канала H.245.

Если канал установления вызова ненадёжный, то для обмена сигнализацией применяется протокол H.225.0 [82], используя некоммутируемый способ передачи данных с помощью протокола UDP. В этом случае для установления вызова определён отдельный механизм подтверждения приёма и повторной передачи, т.к. для сигнализации, связанной с установлением вызова, требуется надёжная передача.

Протокол H.225.0 представляет собой протокол сигнализации для установления и разъединения H.323 вызова между двумя H.323 оконечными точками. В рамках этого протокола определена и процедура ускоренного соединения (Fast Connect Procedure).

Протокол H.225.0 в рамках процедур, требуемых для установления и разъединения вызова, определяет использование следующих сообщений:

— Setup — сообщение о начале установления соединения;
— Setup Acknowledge — подтверждение установления соединения;
— Information — информация, необходимая для установления вызова, или другие сведения, относящиеся к вызову;
— Call Proceeding — сообщение о продолжении установления вызова;
— Progress — в этом сообщении посылается информация о дальнейшем развитии вызова при взаимодействии с сетями с коммутацией каналов;
— Alerting — оповещение о входящем вызове;
— Connect — сообщение об установлении соединения;

— Facility — сообщение для осуществления дополнительных услуг и туннелирования H.245-сообщений по каналам установления вызова;
— Status Inquiry — запрос статуса вызова;
— Status — сообщение содержит статус вызова из аспекта отправителя сообщения и причину его передачи;
— Notify — уведомление, содержащее информацию о вызове, например, индикацию временного прерывания (user suspend) или возобновления (user resume) вызова;
— Release Complete — полное разъединение вызова.

Протокол сигнализации RAS (Registration, Admission and Status — регистрация, подтверждение и статус) применяется для передачи служебных сообщений между терминалами и контроллером зоны. RAS-сообщения служат для регистрации терминалов, допуска их к сеансу связи, изменения используемой полосы пропускания, информирования о состоянии сеанса и его прекращении. В отсутствие контроллера зоны протокол RAS не используется.

Основные процедуры в рамках протокола RAS:
— обнаружение контроллера зоны;
— регистрация оконечного узла (терминала);
— управление доступом;
— управление шириной полосы пропускания;
— определение местонахождения оконечного узла;
— получения подробной статусной информации о вызовах.

Оконечные узлы используют протокол RAS для обнаружения контроллеров зоны, регистрации, а затем для получения разрешения на право использования части ресурсов системы, а также для получения транспортных адресов других удалённых оконечных узлов. Контроллеры зоны, в свою очередь, посредством процедур регистрации и одобрения доступа используют протокол RAS для управления своей зоной, надзора за статусом регистрированных оконечных узлов, управления шириной полосы пропускания и определения местоположения оконечных узлов в других зонах посредством обмена адресной информацией с их контроллерами зоны.

9.2.8. Мультимедийная передача.

Протокол RTP (RFC 1889) обеспечивает в IP-сетях доставку адресатам аудио- и видеопотоков в масштабе реального времени. RTP идентифицирует тип и номер пакета, устанавливает в него метку синхронизации. На основе этой информации приёмный терминал синхронизирует звук, видео и данные, осуществляет их последовательное и непрерывное воспроизведение. Корректное функционирование RTP возможно при наличии в абонентских терминалах механизмов буферизации принимаемой информации.

Транспортный протокол управления передачей в режиме реального времени RTCP (RFC 1889) контролирует реализацию функций RTP. Он также отслеживает качество обслуживания и снабжает соответствующей информацией компоненты, участвующие в конференции.

9.2.9. Эволюция H.323

Первоначально протокол H.323 был предназначен исключительно для локальных сетей и не охватывал проблемы QoS и надёжности.

Вторая версия протокола H.323 одобрена в феврале 1998 г. Были введены некоторые новые функции, связанные с технологией VoIP.

Особое внимание уделялось механизмам обеспечения надёжной H.323-коммуникации в рамках рекомендации H.235 [83]:

- подтверждение достоверности — механизм, которым подтверждается достоверность оконечных точек, участвующих в конференции;
- неприкосновенность данных — механизм контроля целостности принятых пакетных данных;
- защита персональной информации / конфиденциальность коммуникации путём кодирования и декодирования;
- невозможность отрицания — способ предотвращения возможности отрицания участия в конференции.

Другим улучшением в этой версии стало добавление процедуры ускоренного соединения (Fast Connect) — нового быстрого метода установления вызова, а также процедуры передачи сообщений H.245 по каналу установления вызова (туннелирование), вследствие чего потребность надёжных (TCP) соединений в вызове была сведена к одному соединению, что дополнительно сократило время установления вызова.

Кроме того, во второй версии H.323 были введены первые дополнительные услуги в серии рекомендаций H.450: переадресация вызова (Call Transfer) [84] и изменение маршрута вызова (Call Diversion) [85].

Вторая версия охватывает и некоторые аспекты QoS, обеспечивая оконечным точкам H.323 возможность введения параметров качества для медиа потоков.

Улучшения второй версии коснулись и контроллера зоны — была введена концепция альтернативных контроллеров зоны, на которые перенаправлялась часть запросов с основного контроллера зоны в случае высокой нагрузки или неисправности.

Третья версия была утверждена в сентябре 1999 г. В этой версии стала возможной передача сигнализации для установления большего числа вызовов посредством одного TCP-соединения, а также удержание установленного TCP-соединения при отсутствии вызова.

Дополнение E (Annex E) к рекомендации H.323 обеспечило стандарту H.323 альтернативное решение — передачу сигнализации для установления вызова по ненадёжному каналу, используя протокол UDP, что позволило уменьшить время установления вызова и улучшить управление такими параметрами, как время повторной передачи неподтверждённых сообщений и определение ошибки удалённой H.323 оконечной точки, с которой осуществлена коммуникация.

В третьей версии было разработано дополнение рекомендации H.225.0, Annex G, описывающее методы и сигнализацию, необходимую для распознавания адреса, разрешения доступа, обмена информацией о тарификации и ценах на услуги, а также регистрации использования между административными доменами. Кроме того, это дополнение ввело в архитектуру H.323 новый элемент — разграничитель (Border Element).

В четвёртую версию, принятую в ноябре 2000 г., было введено множество улучшений с целью удержания в то время передовой позиции протокола VoIP. Улучшения коснулись надёжности, наращиваемости и гибкости структуры H.323.

Протокол для взаимодействия межсетевого шлюза (Media Gateway, MG) и контроллера межсетевого шлюза (Media Gateway Controller, MGC) разработан исследовательской группой 16 организации ITU- T в сотрудничестве с организацией IETF и описан в рекомендации H.248.

Помимо того, что в четвёртой версии была расширена группа поддерживаемых дополнительных услуг, также были введены два новых механизма предоставления дополнительных услуг — механизм управления H.323-устройствами, базирующийся на протоколе HTTP, и механизм управления, базирующийся на стимулировании. Механизмы описаны в дополнениях (Annex K и L) к рекомендации H.323.

Пятая версия H.323 была одобрена в июле 2003 г. и, в отличие от предыдущих версий, была направлена на стабилизацию протокола.

9.3. Система сигнализации №7

Система общеканальной сигнализации № 7 (ОКС7 — Signaling System № 7, SS7) обеспечивает связь между коммутационными станциями и специализированными узлами сетей связи.

Система применяется в следующих сетях связи:

- в телефонных сетях общего пользования (Public Switched Telephone Network, PSTN);
- в цифровых сетях с интеграцией служб (Integrated Services Digital Network, ISDN);
- в подвижных сетях (Public Land Mobile Network, PLMN),
- в сети сотовой подвижной связи стандарта GSM (Global System for Mobile communications);
- при реализации концепции интеллектуальной сети (Intelligent Network, IN).

9.3.1. Архитектура сети SS7

В сети SS7 определены следующие базовые функциональные элементы:

- *пункт сигнализации (Signaling Point, SP)* — любой узел сигнальной сети, реализующий функции обработки сигнальных сообщений SS7;
- *звено сигнализации (Signaling Link, SL)* — канал передачи данных, соединяющий между собой пункты сигнализации и состоящий из физического канала связи, терминального сигнального оборудования и протокола, контролирующего соединение;
- *транзитный пункт сигнализации (Signaling Transfer Point, STP)* — пункт сигнализации, осуществляющий только функции маршрутизации сигнальных сообщений между различными звеньями сигнализации и не имеющий подсистем пользователей.

Архитектура SS7 имеет модульную структуру, состоящую из нескольких функциональных блоков, причём в основе архитектуры лежит принцип разделения функций между *подсистемой передачи сообщений (Message Transfer Part, MTP)* и *подсистемами пользователей сигнальной сети (User Parts, UP)*. Подсистема передачи сообщений поддерживает не все необходимые функции маршрутизации и адресации сообщений, предусмотренные в модели ISO/OSI на сетевом уровне. Поэтому в SS7 введена подсистема управления сигнальным соединением (Signaling Connection Control Part, SCCP), предоставляющая расширенные услуги по адресации и передаче сообщений.

9.3.1.1. Подсистема MTP

Подсистема MTP обеспечивает корректную передачу информации между узлами сети сигнализации.

Подсистема состоит из следующих элементов:

— *звена передачи данных (MTP1)*;
— *функций управления звеном сигнализации (MTP2)*;
— *функций управления сетью сигнализации (MTP3)*.

MTP1 представляет собой полнодуплексное физическое соединение, состоящее из двух физических каналов, передающих информацию в противоположных направлениях с одинаковой скоростью.

MTP2 вместе с MTP1 образуют *звено сигнализации*, которое обеспечивает достоверную передачу сигнальных сообщений между двумя смежными пунктами сигнализации.

Базовыми элементами звена сигнализации являются *сигнальные единицы (Signal Unit)* — блоки данных переменной длины, в которых передаются любые другие сообщения SS7. Используется три типа сигнальных единиц:

— *значащие сигнальные единицы (Message Signal Unit, MSU)* — содержат данные подсистем пользователей или управляющую информацию MTP3;
— *сигнальные единицы состояния звена (Link Status Signal Unit, LSSU)* — содержат управляющую информацию уровня звена сигнализации;
— *заполняющие сигнальные единицы (Fill-In Signal Unit, FISU)* — генерируются звеном сигнализации, когда нет других сигнальных сообщений, и предназначены для контроля за работоспособностью звена сигнализации, так что дефектные звенья могут быть быстро обнаружены и отключены.

На рис. 9.3, 9.4, 9.5 приведены форматы сигнальных единиц MSU, LSSU, FISU.

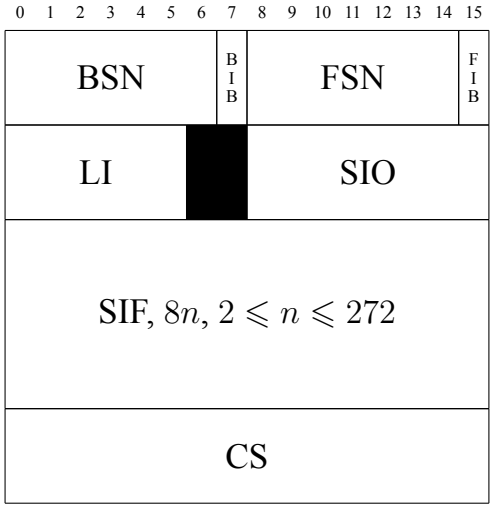


Рис. 9.3. Формат MSU

Все сигнальные единицы начинаются 8-битным полем *Флаг (Flag)*.

Поле *Обратный порядковый номер (Backward Sequence Number, BSN)* (длина 7 бит) содержит номер подтверждаемой сигнальной единицы.

Поле *Обратный бит-индикатор (Backward Indicator Bit, BIB)* (длина 1 бит) используется базовым методом коррекции ошибок.

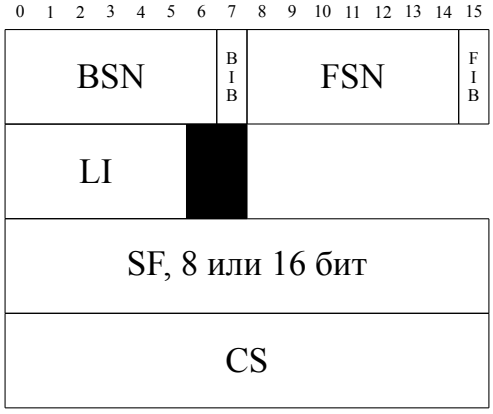


Рис. 9.4. Формат LSSU

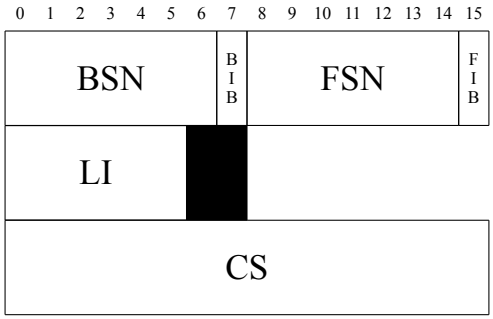


Рис. 9.5. Формат FISU

Поле *Прямой порядковый номер (Forward Sequence Number, FSN)* (длина 7 бит) содержит номер передаваемой сигнальной единицы.

Поле *Прямой бит-индикатор (Forward Indicator Bit, FIB)* (длина 1 бит) используется базовым методом коррекции ошибок.

Поле *Индикатор длины (Length Indicator, LI)* (длина 6 бит) характеризует тип сигнальной единицы: LI > 2 соответствует MSU, LI = 1 или 2 соответствует LSSU, LI = 0 соответствует FISU.

Поле *Байт служебной информации (Service Information Octet, SIO)* (длина 8 бит) состоит из индикатора службы (Service Indicator, SI), используемого в MSU для привязки сигнальной информации к конкретной подсистеме пользователя, и поля подслужбы (Sub-service Field), содержащего *индикатор сети (Network Indicator, NI)*.

Поле сигнальной информации (Signaling Information Field, SIF) имеет переменную длину и содержит информацию, предоставленную верхними уровнями для передачи. Длина этого поля состоит из целого числа байт, не менее 2 и не более 272.

Поле состояния (Status Field, SF) (длина 1 или 2 байта) передаётся только в составе LSSU и содержит информацию о состоянии звена.

Поле контрольной суммы (Check Sum, CS) (длина 16 бит) используется для контроля целостности.

MTP2 выполняет следующие функции:

— инициализацию звена сигнализации;
— определение границ сигнальных единиц и их первичное кодирование/ де-

кодирование;
— управление потоком;
— индикацию перегрузок для MTP3;
— обработку отказа управляющего процессора;
— обнаружение ошибок;
— коррекцию ошибок;
— мониторинг ошибок.

MTP3 содержит функции и процедуры управления передачей информации между различными пунктами в сети сигнализации, которые являются общими для всех звеньев сигнализации, но не зависят от их функционирования по отдельности. При помощи MTP3 осуществляются обработка сигнальных сообщений и управление сетью сигнализации.

Функции обработки сигнальных сообщений гарантируют доставку сигнальных единиц от подсистемы пользователя MTP, находящейся в *исходящем пункте сигнализации (Originating Point, OP)*, к такой же подсистеме, находящейся в *пункте назначения (Destination Point, DP)*. Эти функции базируются на поле NI в байте служебной информации (SIO) и метке маршрутизации, которые содержатся в значащих сигнальных единицах и явно определяют используемое звено сигнализации, исходящий пункт и пункт назначения.

Функции сетевого управления позволяют настраивать и гибко реконфигурировать сигнальную сеть в случае сбоев в звеньях или пунктах сигнализации, а также осуществлять контроль за сигнальным трафиком для предотвращения перегрузок.

9.3.1.2. Подсистема SCCP

Подсистема SCCP для реализации функций адресации сообщений помимо *кода пункта назначения DPC (Destination Point Code)* использует *номер подсистемы (Subsystem Number, SSN)*, который идентифицирует подсистему пользователя SCCP внутри узла сети. Кроме того, SCCP позволяет адресовать сообщения с помощью *глобальных наименований (Global Titles, GT)*, в качестве которых могут выступать, например, цифры телефонного номера абонента.

В SCCP определены следующие услуги расширенной передачи сообщений:
— простая передача без установления соединения — информация пользователя группируется в *блоки данных (Network Service Data Unit, NSDU)*, которые доставляются функциями SCCP в пункт назначения независимо друг от друга и без учёта последовательности;
— последовательная передача без установления соединения — информация пользователя группируется в NSDU, которые доставляются функциями SCCP в пункт назначения независимо друг от друга, но с учётом последовательности;
— простая передача с установлением соединения — двунаправленная передача NSDU через установленное сигнальное соединение с возможностью обеспечения сохранения последовательности, сегментации и сборки сообщений для передачи блоков NSDU длиной более 255 байт;
— установление соединения с контролем потока данных — двунаправленная передача NSDU через установленное сигнальное соединение с возможностью управления потоком, а также обнаружением потери и нарушения последовательности сообщений.

В SCCP входят следующие функциональные блоки:

— *блок функций передачи, ориентированной на соединение (SCCP Connection-Oriented Control, SCOC)*, — контролирует установление, функционирование и разъединение сигнальных соединений;
— *блок функций передачи, не ориентированной на соединение (SCCP Connectionless Control, SCLC)*, — осуществляет передачу NSDU без установления соединения;
— *блок управления (SCCP Management, SCMG)* — осуществляет контроль за сбоями и перегрузками, возникающими в подсистемах пользователей SCCP или сигнальных маршрутах, позволяет перенаправлять сигнальные сообщения резервным подсистемам пользователя в случае недоступности основной подсистемы;
— *блок маршрутизации (SCCP Routing Control, SCRC)* — осуществляет маршрутизацию сообщений.

9.3.1.3. Подсистемы UP

Подсистемы UP генерируют и обрабатывают сигнальные сообщения, а также реализуют функции, специфические для конкретных типов UP сигнальной сети.

В зависимости от сети связи, для которой применяется система SS7, выделены следующие подсистемы UP:
— подсистема пользователя телефонии (Telephone User Part, TUP);
— подсистема пользователя сети ISDN (ISDN User Part, ISUP);
— прикладная подсистема обеспечения транзакций (Transaction Capabilities Application Part, TCAP);
— подсистема пользователя подвижной связи (Mobile Application Part, MAP);
— подсистема пользователя интеллектуальной сети (Intelligent Network Application Protocol, INAP).

9.3.1.3.1. Подсистема TUP. Подсистема TUP посредством обмена сообщениями обеспечивает поддержку функций управления телефонными вызовами в национальных и международных сетях. Сообщение содержит метку, код заголовка, а также один или несколько сигналов и/или индикаторов.

Метка состоит из следующих полей:
— *код точки назначения (Destination Point Code, DPC)* — указывает точку в системе сигнализации, для которой предназначено данное сообщение;
— *код точки отправления (Originating Point Code, OPC)* — указывает точку в системе сигнализации, отправившую данное сообщение;
— *код идентификации канала (Circuit Identification Code, CIC)* — указывает голосовой канал, непосредственно соединяющий точки назначения и отправления.

Код заголовка состоит из двух частей — H0 и H1. H0 идентифицирует указанную группу сообщений, а H1 содержит код сигнализации или (для более сложных сообщений) идентифицирует формат этих сообщений.

9.3.1.3.2. Подсистема ISUP. Подсистема ISUP определяет функции и процедуры передачи и обработки межстанционных сигналов для обеспечения служб коммутации каналов и возможностей пользователя сети ISDN.

Взаимодействие с пользователем ISDN осуществляется по протоколу управления вызовом (Call Control) Q.931 [64]. Для установления, управления и разъединения ISDN-соединений используются сообщения ISUP (рис. 9.6), передаваемые между коммутационными станциями и транзитными пунктами сигнализации.

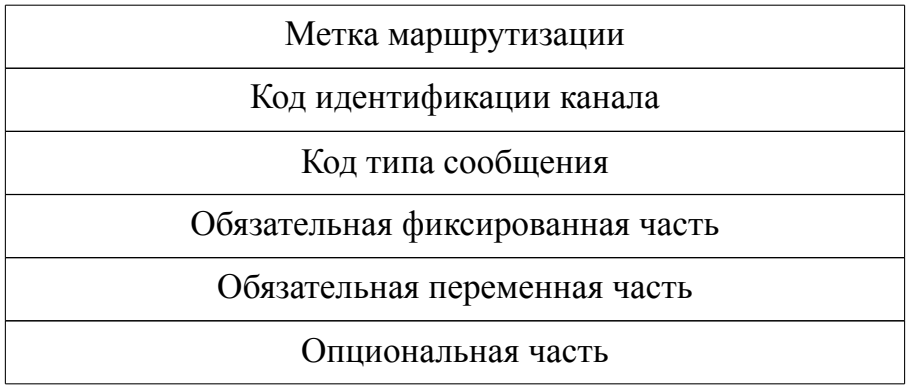


Рис. 9.6. Формат сообщений ISUP

Поле *Метка маршрутизации (Routing Label)* является частью заголовка MTP, одинаковой для всех сообщений данного соединения.

Поле *Код идентификации канала (Circuit Identification Code, CIC)* идентифицирует канал, по которому передаётся сообщение.

Поле *Код типа сообщения (Message Type Code)* идентифицирует сообщение ISUP.

Поле *Обязательная фиксированная часть (Mandatory Fixed Part)* содержит обязательные, имеющие постоянную длину параметры для определённого типа сообщения.

Поле *Обязательная переменная часть (Mandatory Variable Part)* содержит обязательные параметры с переменной длиной.

Поле *Опциональная часть (Optional Part)* содержит необязательные параметры.

Сообщения ISUP делятся на пять категорий:

— сообщения установления соединения в прямом направлении (Forward Setup):
  – *первоначальный адрес сообщения (Initial Address Message, IAM)* — передаётся в прямом направлении для инициации занятия исходящего канала, передачи адреса и относящейся к нему информации;
  – *последующий адрес сообщения (Subsequent Address Message, SAM)* — может передаваться после сообщения IAM для передачи дополнительной информации от вызывающей стороны;
— общие сообщения установления соединения (General Setup):
  – *информационный запрос (Information Request)* — запрос дополнительной информации, относящейся к вызову;
  – *информация (Information)* — передача дополнительной информации, относящейся к вызову;
  – *сообщение продолжения (Continuity)* — передаётся в прямом направлении для индикации продолжения подключения информационного канала к следующей станции;

— сообщения установления соединения, передаваемые в обратном направлении (Backward Setup):
  – *адрес полученного сообщения (Address Complete Message, ACM)* — передаётся в обратном направлении и инициирует получение станцией назначения всей необходимой адресной информации для направления вызова вызываемой стороне;
  – *соединение (Connect)* — посылается в обратном направлении для индикации того, что вся адресная информация, необходимая для направления вызова вызываемой стороне, получена и на вызов дан ответ;
  – *прогрессивный вызов (Call Progress)* — сообщает, что событие, произошедшее в течение установления соединения, должно было относиться к вызывающей стороне;
— сообщения управления вызовом (Call Supervision):
  – *ответ (Answer)* — посылается в обратном направлении для индикации того, что на вызов получен ответ;
  – *переадресация вызова (Forward Transfer)* — сигнализирует о необходимости переадресации исходящего вызова;
  – *отключение (Release)* — сообщает, что канал, определённый сообщением, освобождён;
— сообщения управления каналом (Circuit Supervision):
  – *подтверждение отключения (Release Complete)* — передаётся в ответ на сообщение Release после освобождения канала;
  – *проверка целостности (Continuity Check Request)* — передаётся станцией, для канала которой осуществляется проверка целостности, к станции на другом конце канала, с запросом подсоединения оборудования проверки целостности;
  – *освобождение канала (Reset Circuit)* — передаётся для освобождения канала, когда из-за неисправности отсутствует сообщение Release или Release Complete;
  – *блокировка (Blocking)* — передаётся для целей техобслуживания к станции на другом конце канала, в результате чего происходит блокировка инициализации последовательных исходящих вызовов по этому каналу;
  – *разблокировка (Unblocking)* — передаётся к станции на другом конце канала для отмены состояния блокировки канала, обусловленного предварительно посланным сообщением Blocking или Group Blocking;
  – *подтверждение блокировки (Blocking Acknowledgment)* — подтверждает получение сообщения Blocking;
  – *подтверждение разблокировки (Unblocking Acknowledgment)* — подтверждает получение сообщения Unblocking;
  – *сообщение приостановления (Suspend)* — используется для индикации временного отключения абонентского терминала;
  – *сообщение возобновления соединения (Resume)* — сообщение о восстановлении соединения после Suspend;

— сообщения управления группой каналов (Circuit Group Supervision):

  – *блокирование группы каналов (Circuit Group Blocking)* — передаётся для целей техобслуживания к станции на другом конце группы каналов, в результате чего происходит блокировка последовательных исходящих вызовов по каналам этой группы;

  – *разблокирование группы каналов (Circuit Group Unblocking)* — передаётся к станции на другом конце группы каналов для отмены состояния блокировки группы каналов, обусловленного предварительно посланным сообщением Circuit Group Blocking;

  – *подтверждение блокировки группы каналов (Circuit Group Blocking Acknowledgment)* — подтверждает получение сообщения Circuit Group Blocking;

  – *подтверждение разблокировки группы каналов (Circuit Group Unblocking Acknowledgment)* — подтверждает получение сообщения Circuit Group Unblocking;

  – *перезапуск группы каналов (Circuit Group Reset)* — передаётся для освобождения группы каналов, когда из-за неисправности невозможно определить, каким каналам соответствуют сигналы освобождения;

  – *подтверждение перезапуска группы каналов (Circuit Group Reset Acknowledgment)* — передаётся в ответ на сообщение перезапуска группы каналов.

9.3.1.3.3. Подсистема TCAP. Подсистема TCAP предназначена для взаимодействия сетевых приложений. В сетях связи распределённые приложения, использующие TCAP, обычно функционируют на коммутационных станциях и в сетевых базах данных. Основной функцией TCAP в этих сетях является вызов удалённых процедур для поддержки услуг интеллектуальной сети.

TCAP разделена на два подуровня: *подуровень компонент (Component Sublayer, CSL)* и *подуровень транзакций (Transaction Sublayer, TSL)*. Подуровень компонент осуществляет обмен компонент (запрос на действие на удалённом конце или ответ на вызванную операцию) между транзакциями пользователей. Подуровень транзакций ответственен за обмен сообщениями, которые содержат эти компоненты.

Для TCAP определены следующие типы сообщений:

— безадресное сообщение (Undirectional);
— сообщение начала (Begin);
— сообщение продолжения (Continue);
— сообщение окончания (End);
— сообщение прерывания (Abort).

Для TCAP определены следующие типы компонент:

— вызов (Invoke);
— возвращение результата (Return Result);
— возвращение ошибки (Return Error);
— отказ (Reject).

9.3.1.3.4. Подсистема MAP. Подсистема MAP обеспечивает выполнение процедур сигнализации, необходимых для обмена информацией в сетях сотовой подвижной связи, например стандарта GSM.

Сообщения MAP передаются между мобильными коммутаторами и базами данных для поддержки аутентификации пользователей, идентификации оборудования и роуминга. Передача сообщений осуществляется с помощью протокола TCAP.

9.3.2. Преимущества и недостатки SS7

Система SS7 обладает следующими преимуществами:

— оптимизирована для работы в цифровых сетях связи в сочетании со станциями с программным управлением (Stored Program Control), но может работать и в аналоговых сетях на скорости до 64Кбит/с;
— обеспечивает надёжные средства передачи информации в правильной последовательности, без потерь или дублирования;
— имеет чёткую функциональную архитектуру, которая обеспечивает гибкость и модульность для различных применений при сохранении единой концепции системы;
— может использовать наземные и спутниковые физические каналы передачи данных.

9.4. Концепция Softswitch

Softswitch является носителем интеллектуальных возможностей сети, который координирует управление обслуживанием вызовов, сигнализацию и функции, обеспечивающие установление соединения через одну или несколько сетей.

Softswitch:

— управляет обслуживанием вызовов;
— координирует обмен сигнальными сообщениями между сетями.

9.4.1. Архитектура Softswitch

Архитектура Softswitch (рис. 9.7) представляет собой набор функциональных объектов (функций, а не физических объектов), соединённых между собой посредством интерфейсов.

В зависимости от своей функциональности функциональные объекты (ФО) распределены по функциональным уровням. Выделяют четыре функциональных уровня:

— *транспортный уровень (Transport Plane)* — отвечает за транспортировку сообщений по сети связи и обеспечивает доступ к сети IP-телефонии сигнальной и/или пользовательской информации, поступающей со стороны других сетей или терминалов;
— *уровень управления обслуживанием вызова и сигнализации (Call Control & Signaling Plane)* — управляет основными элементами сети IP-телефонии;
— *уровень услуг и приложений (Service & Application Plane)* — реализует управление услугами и/или приложениями в сети IP-телефонии, их логику и выполнение, а также управление специализированными компонентами передачи пользовательской информации (например, медиасерверами);

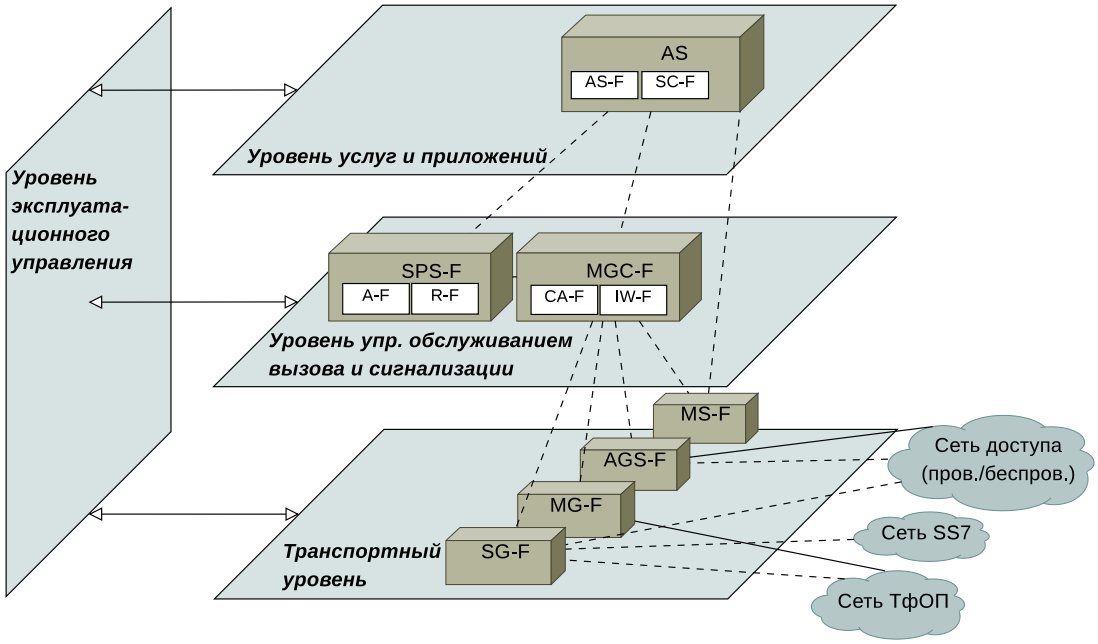


Рис. 9.7. Архитектура Softswitch (пунктирная линия — сигнализация, сплошная — данные)

— *уровень эксплуатационного управления (Management Plane)* — обеспечивает выполнение функций активизации абонентов и услуг, техобслуживания, биллинга и пр.

Элементы транспортного уровня:

— *ФО шлюза сигнализации (Signaling Gateway Function, SG-F)* — обеспечивает обмен сигнальной информацией между сетью IP-телефонии и ТфОП или между транзитной пакетной IP-сетью и сетью сотовой подвижной связи с коммутацией каналов на базе стека SS7; использует протоколы Sigtran типов TUA, SUA и M3UA over SCTP;

— *ФО сигнализации шлюза доступа (Access Gateway Signaling Function, AGS-F)* — обеспечивает обмен сигнальной информацией между сетью IP-телефонии и сетью доступа с коммутацией каналов на базе интерфейса V5.1/V5.2 или ISDN, а также между транзитной сетью подвижной связи с коммутацией пакетов и сетью сотовой подвижной связи на базе TDM или ATM; использует протоколы Sigtran типов TUA, SUA и M3UA over SCTP;

— *ФО медиашлюза (Media Gateway Function, MG-F)* — обеспечивает сопряжение IP-сети с портом доступа, с соединительной линией или с совокупностью портов и/или соединительных линий, выполняя таким образом функции шлюза между пакетной сетью и внешними сетями с коммутацией каналов, такими как ТфОП, сеть сотовой подвижной связи или ATM; использует протоколы и технологии RTP/RTCP, TDM, H. 248 и MGCP;

— *ФО медиасервера (Media Server Function, MS-F)* — обеспечивает управление обработкой пользовательского пакетного трафика от приложений; использует протоколы SIP, MGCP и H. 248.

Элементы уровня управления обслуживанием вызова и сигнализации:

— *ФО контроллера медиашлюзов (Media Gateway Controller Function,*

MGC-F) — представляет собой логический элемент управления обслуживанием вызова и сигнализации для одного или более транспортных шлюзов:

  - *ФО устройства управления шлюзом (Call Agent Function, CA-F)* — обеспечивает обработку вызова и определяет состояние процесса его обслуживания; может использовать протоколы SIP, SIP-T, BICC, H. 323, Q. 931, Q. SIG, INAP, ISUP, TCAP, BSSAP, RANAP, MAP и CAP;
  - *ФО взаимодействия (Interworking Function, IW-F)* — обеспечивает взаимодействие между разными сетями сигнализации (например, IP и ATM, ОКС7 и SIP/H.323 и т. п.);

— *ФО SIP-прокси-сервера (SIP Proxy Server Function, SPS-F)*:

  - *ФО маршрутизации (Routing Function, R-F)* — предоставляет информацию о маршрутизации вызова ФО MGC-F; может использовать протоколы ENUM и TRIP;
  - *ФО учёта стоимости (Accounting Function, A-F)* — собирает учётную информацию о вызовах для целей биллинга, а также обеспечивает аутентификацию, идентификацию и учёт в удалённых сетях; может использовать протоколы RADIUS и AuC.

Элементы уровня услуг и приложений:

— *ФО сервера приложений (Application Server Function, AS-F)* — обеспечивает выполнение услуг для одного или более приложений; использует протоколы SIP, MGCP, H. 248, LDAP, HTTP, CPL и XML;

— *ФО управления услугами (Service Control Function, SC-F)* — обеспечивает управление логикой услуг; использует протоколы INAP, CAP и MAR, открытые API типа JAIN и Parlay.

Физически элементы сети могут выполнять как одну, так и несколько функций, а также одна функция может быть распределена между несколькими элементами сети. Так, в модуле контроллера медиашлюзов могут быть реализованы MGC-F, CA-F, IW-F, R-F/A-F, SPS-F и др.

9.4.2. Протоколы в сетях Softswitch

9.4.2.1. Протокол MGCP

Протокол управления медиашлюзом (Media Gateway Control Protocol, MGCP) является внутренним протоколом для обмена информацией между функциональными блоками распределённого шлюза. Перенос сообщений протокола MGCP обеспечивает протокол UDP.

Для описания процесса обслуживания вызова с использованием протокола MGCP разработана модель организации соединения, в основу которой положены два компонента: *оконечная точка или устройство (Endpoints)* и *подключение (Connections)*.

Оконечная точка — это порт оборудования, являющегося источником или приёмником информации. Порт может быть физическим или виртуальным. Каждый порт определяется идентификатором, содержащим доменное имя шлюза и локальное имя в шлюзе.

Соединение — подключение порта к одному из двух концов соединения, которое создаётся между ним и другим портом. Соединение может связывать порты разных шлюзов через сеть с IP-маршрутизацией или порты внутри одного шлюза.

При установлении, поддержании и разрушении соединения устройство управления и шлюз обмениваются командами и ответами, которые представляют собой набор текстовых строк.

Команды состоят из следующих компонент: *кода команды, идентификатора транзакции, идентификатора порта, версии протокола*.

В протоколе MGCP определены следующие команды:

- CreateConnection (CRCX) — создать соединение;
- ModifyConnection (MDCX) — модифицировать соединение;
- DeledeConnection (DLCX) — завершить соединение;
- Notify (NTFY) — уведомить;
- NotificationRequest (RQNT) — запрос уведомления;
- EndpointConfiguration (EPCF) — конфигурация портов;
- AuditEndpoint (AUEP) — проверить порт;
- AuditConnection (AUCX) — проверить соединение;
- ReStartInPrgress (RSIP) — рестарт.

Ответы состоят из следующих компонент: *кода ответа, идентификатора транзакции, комментария, параметров* (обязательных и не обязательных).

Определены следующие *основные параметры*:

- CallId (C) — идентификатор сеанса связи;
- ConnectionId (I) — идентификатор подключения;
- Mode (M) — режим соединения;
- RequestedInfo (F) — запрашиваемая информация;
- ResponseAck (K) — подтверждение транзакции;
- BearerInformation (B) — закон кодирования;
- RequestIdentifier (X) — идентификатор запроса;
- LocalConnectionOptions (L) — параметры порта;
- RequestedEvents (R) — запрашиваемые события;
- SignalRequests (S) — требование передать сигнал;
- NotifiedEntity (N) — уведомляемый объект;
- DigitMap (D) — план нумерации;
- QuarantineHandling (Q) — карантинная обработка;
- DetectEvents (T) — выявляемые события;
- ConnectionParameters (P) — параметры соединения;
- RestartMethod (RM) — метод рестарта;
- ReasonCode (E) — код причины;
- RestartDelay (RD) — задержка рестарта;
- ObservedEvents (O) — обнаруженные события;
- LocalConnectionDescriptor (LCD) — локальные параметры соединения на передающей стороне;
- RemoteConnectionDescriptor (RCD) — удалённые параметры соединения на приёмной стороне.

9.4.2.2. Протокол Megaco/H.248

Для переноса сигнальных сообщений Megaco/H.248 могут использоваться протоколы UDP, TCP, SCTP или технология ATM.

Для описания процесса обслуживания вызова с использованием протокола Megaco разработана модель организации соединения, в основу которой положены два компонента: *порт (Termination)* и *контекст (Context)*.

Порты являются источниками и приёмниками речевой информации и могут быть физическими (аналоговые телефонные интерфейсы оборудования) или виртуальными (существующие только в течение разговорной сессии).

Контекст — это абстрактное представление соединения двух или более портов одного шлюза. Контекст имеет уникальный идентификатор.

При помощи протокола Megaco/H.248 контроллер может изменять свойства портов шлюза. Свойства портов группируются в дескрипторы, которые включаются в команды управления портами.

Megaco/H.248 определяет восемь команд, которые обеспечивают возможность управления и манипулирования контекстами и окончаниями:

- Add — добавить окончание к контексту;
- Modify — изменить свойства окончания;
- Subtract — удалить окончание из контекста;
- Move — переместить окончание из одного контекста в другой;
- AuditValue — определить текущее состояние окончания;
- AuditCapabilities — определить состояния, которые может принимать окончание;
- Notify — уведомить о событиях, которые произошли в транспортном шлюзе;
- ServiceChange —уведомить об изменении обслуживания.

Megaco/H.248 определяет ряд дескрипторов, предназначенных для использования вместе с командами и ответами:

- дескриптор модема — специфицирует тип модема и связанные с ним параметры, которые следует использовать в соединениях модема при передаче аудио, видео или данных;
- дескриптор мультиплексирования — характеризует тип мультиплексирования в мультимедийном терминале;
- дескриптор среды — описывает различные информационные потоки (медиапотоки);
- дескрипторы потока — используются между MG и Softswitch для указания, какие медиапотоки взаимосвязаны;
- дескриптор среды — описывает различные информационные потоки (медиапотоки);
- дескрипторы LocalDescriptor и RemoteDescriptor — содержат или не содержат несколько описаний сеансов SDR, определяющих сеанс на локальном и удалённом концах соединения соответственно;
- дескриптор событий — содержит Requestldentifier и список событий, которые MG должен обнаруживать;
- дескриптор сигналов — содержит список сигналов, которые должно подавать оконечное оборудование;
- дескриптор проверки — задаёт перечень информации, которую необходимо передавать из MG в Softswitch;
- дескриптор ServiceChangeDescriptor — используется только в сочетании с командой ServiceChange и включает в себя тип изменения обслуживания, причину изменения обслуживания и новый адрес для использования после изменения обслуживания;
- дескриптор DigitMap — описывает план нумерации;
- дескриптор StatisticsDescriptor — содержит информацию, которая относится к использованию оконечного оборудования в данном контексте;
- дескриптор ObservedEvents — используется для информирования Softswitch об обнаруженных событиях;

— дескриптор Error — передаётся в ответе, когда не может быть выполнена команда.

Команды могут группироваться в *транзакции*, причём в одной транзакции могут быть команды, относящиеся к разным контекстам. После приёма транзакции получатель последовательно выполняет команды, вложенные в неё.

Несколько транзакций могут передаваться по сети в виде *сообщений*, снабжённых заголовком, идентифицирующим отправителя. *Идентификатором сообщения (Message Identifier, MID)* служит назначенное имя (например, адрес в домене, имя в домене, имя устройства) объекта, передающего сообщение. Транзакции в пределах сообщения обрабатываются в произвольном порядке. Сообщения Megaco/H.248 по сути являются только транспортным механизмом.

Протокол Megaco/H.248 определяет типовые наборы характеристик, сигналов и событий для Softswitch и шлюзов разных типов, чтобы обеспечить возможность их взаимодействия. Типовой набор характеризуется базовым описанием, свойствами, предусматриваемыми событиями, поддерживаемыми сигналами, предоставляемыми статистическими данными, любыми процедурами, относящимися к надлежащей поддержке набора. Он содержит следующие разделы:

— *Package* — содержит общее описание набора, определяющее его имя, идентификатор, текстовое описание, версию и опциональные поля;
— *Properties* — определяет свойства (характеристики) набора и содержит имя каждого свойства, его идентификатор, текстовое описание, тип, возможные значения, специфицирующие свойство и характеристики;
— *Events* — определяет событие и содержит имя события, его идентификатор, текстовое описание, параметры дескриптора Events и параметры дескриптора ObservedEvents;
— *Signals* — определяет сигналы, имя и идентификатор каждого сигнала, его текстовое описание, тип, продолжительность, дополнительные параметры;
— *Statistics* — определяет статистические данные, содержит имя и идентификатор данных каждого вида, их текстовое описание, единицы измерения;
— *Procedures* определяют дополнительные аспекты использования набора.

9.4.2.3. Протокол SIP

Протокол инициирования сеансов (Session Initiation Protocol, SIP) разработан группой MMUSIC (Multiparty Multimedia Session Control) комитета IETF [86, 87] и используется для организации, модификации и завершения сеансов связи. Протокол SIP не принимает непосредственного участия в передаче голосовых, видео и других данных, а лишь отвечает за установление связи.

В основу протокола рабочая группа MMUSIC заложила следующие принципы:

— персональная мобильность пользователей — услуги связи предоставляются вне зависимости от местонахождения пользователя;
— масштабируемость сети;
— расширяемость протокола — возможно дополнение протокола новыми функциями при введении новых услуг и его адаптации к работе с различными приложениями.

Кроме того, протокол SIP поддерживает преобразование имён, переадресацию, маршрутизацию, идентификацию и аутентификацию пользователя при его перемещении из одного места в другое.

В сети на базе SIP определены следующие элементы:

— *клиент UAC (User Agent Client)* — инициирует SIP-запросы;
— *сервер UAS (User Agent Server)* — принимает запросы и передаёт обратно ответы:
  – *прокси-сервер (Proxy Server)* — обрабатывает запросы пользователя;
  – *сервер переадресации (Redirect Server)* — предназначен для определения текущего адреса вызываемого пользователя;
  – *сервер регистрации местоположения (Registrars / Location Server)* — позволяют агентам регистрировать своё местоположение, реализуя тем самым услуги мобильности.

Все сообщения SIP делятся на запросы клиента серверу и ответы сервера клиенту. Сообщения SIP могут переноситься как протоколом TCP, так и протоколом UDP. Все сообщения SIP представляют собой последовательности текстовых строк, структура и синтаксис которых соответствуют протоколу HTTP:
— *стартовая строка* — представляет собой начальную строку любого SIP-сообщения и содержит в случае запроса *тип запроса, текущий адрес узла-адресата, номер версии протокола*, а в случае ответа — *номер версии протокола, тип ответа, короткую расшифровку ответа*;
— *заголовки сообщений* — содержат информацию, необходимую для обработки сообщения:
  – *общие заголовки*: Call-ID (идентификатор соединения), Contact (контакт), CSeq (порядковый номер запроса/ответа), Date (дата), Encryption (кодирование), From (источник запроса), To (адресат), Via (путь), Record-Route (запись маршрута);
  – *заголовки содержания* — переносят информацию о размере тела сообщения или об источнике запроса;
  – *заголовки с дополнительной информацией о запросе*: Accept (принимается), Accept-Encoding (кодирование принимается), Accept-Language (язык поддерживается), Authorization (авторизация), Hide (скрыть), Max-Forwards (максимальное количество переадресаций), Organization (организация), Priority (приоритет), Proxy-Authorization (авторизация прокси-сервера), Proxy-Require (требование прокси-сервера), Route (маршрут), Response-Key (ключ кодирования ответа), Subject (тема), User-Agent (агент пользователя);
  – *заголовки с дополнительной информацией об ответе*: Allow (разрешение), Proxy-Authenticate (подтверждение подлинности прокси-сервера), Retry-After (повторить через некоторое время), Server (сервер), Unsupported (не поддерживается), Warning (предупреждение), WWW-Authenticate (аутентификация WWW-сервера);
— *тело сообщения* — содержит запросы (команды) SIP:
  – *INVITE* — приглашает пользователя принять участие в сеансе связи, и обычно содержит описание сеанса связи, вид принимаемой информации и параметры, необходимые для приёма информации;
  – *ACK* — подтверждает приём ответа на команду INVITE, содержит описание сеанса связи, переданное вызывающим пользователем;
  – *CANCEL* — отменяет обработку ранее переданных запросов;
  – *BYE* — разрушает соединение;

- *REGISTER* — сообщает текущее местоположение пользователя;
- *OPTIONS* — содержит информацию о возможностях терминального оборудования вызываемого пользователя;
- *INFO* — используется для переноса между шлюзами сигнальных сообщений в течение сеанса связи, для переноса сигналов DTMF, созданных в ходе сеанса, для переноса информации об остатке на счете (билинговой информации), для переноса между участниками сеанса связи изображений и другой не потоковой информации;
- *SUBSCRIBE* — подписка на предоставление информации о состоянии определённого ресурса;
- *MESSAGE* — предназначен для реализации служб интерактивного обмена текстовыми сообщениями с использованием модели, аналогичной отправке SMS.

Для организации взаимодействия с существующими приложениями IP-сетей и обеспечения мобильности пользователей протокол SIP использует адрес, подобный адресу электронной почты. В качестве адресов рабочих станций используются SIP URL, которые бывают четырёх типов: `имя@домен`, `имя@IP-адрес`, `имя@хост`, `№телефона@шлюз`. Первая часть адреса идентифицирует пользователя, зарегистрированного в домене или на рабочей станции, а вторая часть — устройство или домен.

9.4.3. Протокол SDP

Протокол описания сессий (Session Description Protocol, SDP) [88] содержит механизм описания характеристик сеанса — время проведения, требуемые ресурсы и т.д. В SDP предусмотрена возможность изменения параметров сеансов в оперативном режиме.

SDP содержит следующие данные:
- информацию о медиапотоках;
- адреса назначения медиапотоков;
- номера UDP портов для отправителя и получателя;
- типы потока;
- медиаформаты, которые могут использоваться во время сессии;
- время начала, завершения и повторов сессии;
- информацию об инициаторе широковещательной сессии.

Описание сессии SDP:
- поле *Версия протокола* содержит версию протокола SDP;
- поле *Владелец / создатель и идентификатор сессии* служит глобальным идентификатором версии описания сессии:
  - username — идентифицирует пользователя;
  - session id — уникальный идентификатор сессии;
  - version — номер версии данного объявления;
  - network type — тип сети (например, «IN» — Интернет);
  - address type — тип адреса (например, «IP4» или «IP6»);
  - address — глобальный уникальный адрес хоста, с которого была создана данная сессия;

- поле *Имя сессии* указывает имя сессии;
- поле *Информация о сессии* может использоваться для определения медиапотока;
- поле *URI описания сессии* указывает на дополнительную информацию о конференции (сессии);
- поле *e-mail адрес*;
- поле *Телефонный номер*;
- поле *Информация о соединении* содержит данные о соединении:
  - network type — тип сети (например, «IN» — Интернет);
  - address type — тип адреса (например, «IP4» или «IP6»);
  - connection address — адрес соединения;
- поле *Информация о ширине полосы пропускания* определяет желаемую ширину полосы пропускания, которая должна использоваться сессией и медиапотоком;
- поле *Время* определяет время начала и конца сессии;
- поле *Интервалы повторения сессий*;
- поле *Объявление временной зоны* определяет сдвиги времени по отношению к базовому времени повторов сессий;
- поле *Криптографический ключ*;
- поле *Атрибуты сессии* могут быть определены как атрибуты «уровня сессии», атрибуты «уровня медиа»;
- поля *Имя медиа* и *Адрес транспорта*:
  - media — содержит тип медиапотока;
  - port — транспортный порт, в который будет передаваться медиапоток;
  - transport — транспортный протокол;
  - fmt list — форматы медиа.

9.4.4. Услуги в сетях Softswitch

Архитектура Softswitch даёт возможность операторам и/или провайдерам услуг предоставлять услуги, реализованные в виде приложений как от производителя Softswitch, так и от сторонних производителей, а также самостоятельно разрабатывать свои собственные приложения. Это возможно благодаря основанным на открытых стандартах прикладным программным интерфейсам API:

- Parlay — платформа для разработки, интеграции и развёртывания приложений на базе технологии Java;
- JAIN (Java Advanced Intelligent Network) — сетевая топология на базе Java, позволяющая осуществлять интеграцию протоколов IP и IN, обеспечивающая переносимость услуг, конвергенцию сетей и защищённый доступ как к телефонным сетям, так и к сетям передачи данных;
- CORBA (Common Object Request Broker Architecture) — открытая, независимая от поставщиков архитектура и инфраструктура, которую используют прикладные вычислительные системы для обеспечения их совместной работы в компьютерных сетях;
- XML (Extensible Markup Language) — язык разметки, который рассматривается как стандартный способ обмена информацией в средах, не использующих общие платформы;

— CPL (Call Processing Language) — язык, который может быть использован для описания и управления услугами IP-телефонии;
— CGI (Common Gateway Interface) — стандарт интерфейса, используемого для связи внешней программы с веб-сервером;
— сервисные Java-приложения.

9.5. Концепция IMS

Концепция *IMS (IP Multimedia Subsystem)* была предложена 3GPP в начале 2003 г. Эта концепция определяет сетевую архитектуру, которая опирается на пакетную транспортную сеть и обеспечивает управление сеансами связи и доставку в рамках этих сеансов любых типов информации — речи, данных, видео, мультимедиа. Следует заметить, что в системах, отвечающих концепции IMS, услуги могут предоставляться разными сервис-провайдерами и доставляться до пользователей по различным (проводным и беспроводным) сетям доступа.

Концепция IMS была стандартизована в спецификациях 3GPP R.5. Позднее к разработке спецификаций и стандартов IMS присоединились другие организации: 3GPP2, занимающаяся разработками для сетей CDMA2000, ETSI, группа Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), работающая в области конвергенции фиксированных сетей. Альянс Open Mobile Alliance (OMA) определил приложения и услуги, работающие поверх IMS, а Internet Engineering Task Force (IETF) — протоколы сетевого уровня. ETSI, отраслевые группы Форума мультисервисной коммутации (Multiservice Switching Forum, MSF) и Альянса для продвижения решений для телекоммуникационной отрасли (Alliance for Telecommunications Industry Solutions, ATIS) одобрили IMS в качестве основы сетевой инфраструктуры следующего поколения.

9.5.1. Архитектура IMS

Архитектура IMS (рис. 9.8) [89] представляет собой набор функций, соединённых стандартными интерфейсами (табл. 9.3). Физически элементы сети могут выполнять как одну, так и несколько функций, а также одна функция может быть распределена между несколькими элементами сети.

Выделяют три уровня:
— *пользовательский уровень или уровень передачи данных (User Plane)* — отвечает за подключение абонентов к инфраструктуре IMS;
— *уровень управления (Control Plane)* — отвечает за все действия по управлению сеансами связи (регистрирует абонентские устройства и направляет сигнальные сообщения протокола SIP к соответствующим серверам приложений);
— *уровень приложений (Application Plane)* — обеспечивает обслуживание конечных пользователей.

Элементы уровня передачи данных:
— *функция обеспечения мультимедийных ресурсов (Media Resource Function, MRF)*:
  – *процессор мультимедийных ресурсов (MRF Processor, MRFP)* — обеспечивает обработку мультимедийных данных;

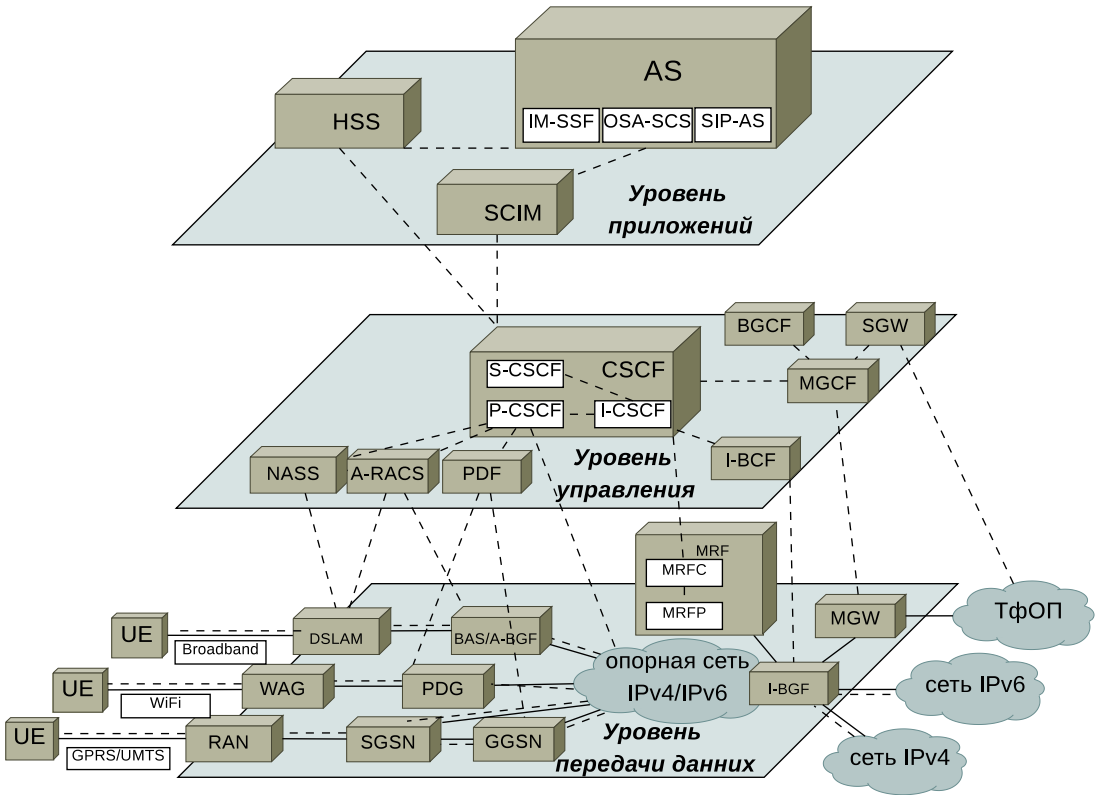


Рис. 9.8. Архитектура IMS (пунктирная линия — сигнализация, сплошная — данные)

– *контроллер мультимедийных ресурсов (MRF Controller, MRFC)* — обеспечивает реализацию услуг конференц-связи, оповещения или перекодирования передаваемого сигнала посредством управления MRFP при помощи протоколов сигнализации;

— *медиа-шлюз (Media Gateway, MGW)* — обеспечивает прямое и обратное преобразование потоков сетей с коммутацией пакетов в потоки сетей с коммутацией каналов;
— *функция межсетевого пограничного шлюза (Interconnect Border Gateway Function, I-BGF)* — обеспечивает взаимодействие между сетями IPv4 и IPv6, отвечает за обеспечение функций безопасности (трансляция адресов и портов NAPT, функции firewall, инструменты QoS);
— *шлюзовой узел GPRS (Gateway GPRS Support Node, GGSN)* — обеспечивает взаимодействие сети сотовой связи и инфраструктуры IMS;
— узел обслуживания абонентов GPRS (Serving GPRS Support Node, SGSN) — обеспечивает обработку данных абонентов GPRS;
— *сети радиодоступа (Radio Access Network, RAN)* — обеспечивают взаимодействие сотовых систем электросвязи и инфраструктуры IMS;
— *шлюз пакетной передачи данных (Packet Data Gateway, PDG)* — обеспечивает доступ пользовательского оборудования WLAN к инфраструктуре IMS, а именно ретранслирует IP-адреса, регистрирует пользовательское оборудование в IMS, обеспечивает выполнение функций безопасности;
— *шлюз беспроводного доступа (Wireless Access Gateway, WAG)* — обеспечивает соединение сетей WLAN и IMS;

— *функция пограничного шлюза доступа для широкополосного пользовательского оборудования (Access Border Gateway Function / Broadband Access Switch, A-BGF/BAS)* — обеспечивает доступ широкополосного пользовательского оборудования к инфраструктуре IMS;
— *цифровой абонентский шлюз доступа (Digital Subscriber Line Access Multiplexer, DSLAM)* — обеспечивает соединение абонентов, использующих широкополосный доступ к инфраструктуре IMS.

Элементы уровня управления:

— *функция управления вызовами и сеансами (Call Session Control Function, CSCF)* — обеспечивает доставку услуг реального времени посредством транспорта IP:
  – *обслуживающая CSCF (Serving CSCF, S-CSCF)* — обрабатывает все SIP-сообщения, которыми обмениваются оконечные устройства;
  – *прокси CSCF (Proxy CSCF, P-CSCF)* — обеспечивает обработку запросов от терминалов IMS к другим элементам IMS, а также выполняет ряд требований, относящихся к обеспечению безопасности (аутентификацию пользователя, контроль за корректностью передаваемых сигнальных сообщений, сбор данных о предоставленных пользователю сервисах);
  – *запрашивающая CSCF (Interrogating CSCF, I-CSCF)* — назначает S-CSCF для конкретного абонента, определяет привилегии абонента по доступу к услугам;
— *функция управления шлюзами (Breakout Gateway Control Function, BGCF)* — управляет маршрутизацией вызовов между сетью с коммутацией каналов (ТфОП или GSM) и сетью IMS;
— *функция управления медиа-шлюзами (Media GatewaysControl Function, MGCF)* — управляет соединениями в транспортных шлюзах IMS, используя H.248/ MEGACO;
— *шлюз сигнализации (Signaling Gateway, SGW)* — обеспечивает преобразование сигнализации ТфОП в вид, понятный MGCF;
— *подсистема управления ресурсами и доступом (Resource and Access Control, RACS)* — обеспечивает функции управления доступом в сеть, управление преобразованием сетевых адресов и портов, присвоение приоритета;
— *функция выбора политики (Policy Decision Function, PDF)* — определяет возможность организации сеанса или его запрета, необходимость изменения параметров сеанса и т.д.;
— *подсистема подключения сети (Network Attachment Subsystem, NASS)* — осуществляет динамическое назначение IP-адресов, аутентификацию на IP-уровне, авторизацию доступа к сети, управление местонахождением на IP-уровне.

Элементы уровня приложений:

— *элемент управления взаимодействием возможных услуг (Service Capability Interaction Manager, SCIM)* — обеспечивает управление взаимодействием плоскости приложений и ядра IMS;
— *SIP-сервер приложений (SIP Application Server, SIP AS)* — обеспечивает выполнение услуг на базе SIP;

— *сервер возможных услуг, базирующихся на открытом доступе к услугам (Open Service Access — Service Capability Server, OSA-SCS)* — обеспечивает доступ к услугам посредством стандартного программного интерфейса приложений;
— *сервер коммутации услуг (IP Multimedia – Service Switching Function, IM-SSF)* — служит для взаимодействия подсистемы IMS с услугами, разработанными для системы мобильной связи GSM;
— *сервер телефонных приложений (Telephony Application Server, TAS)* — принимает и обрабатывает сообщения протокола SIP, обеспечивает базовые сервисы обработки вызовов (включая анализ цифр, маршрутизацию, установление, ожидание и перенаправление вызовов, конференц-связь и т.д.), обеспечивает сервисную логику для обращения к медиасерверам при необходимости воспроизведения оповещений и сигналов прохождения вызова, отвечает за сигнализацию SIP к функции MGCF для выдачи команды медиашлюзам на преобразование битов речевого потока TDM (ТфОП) в поток IP RTP и направление его на IP-адрес соответствующего IP-телефона;
— *сервер домашних абонентов (Home Subscriber Server, HSS)* — обеспечивает открытый доступ в режиме чтения/записи к индивидуальным данным пользователя, связанным с услугами.

Таблица 9.3

Описание стандартных интерфейсов

Название интерфейса	**Элементы IMS**	**Описание**	**Протокол**
Cr	MRFC, AS	Используется MRFC для передачи данных (скриптов и др/ ресурсов) от AS	HTTP поверх TCP/SCTP
Cx	I-CSCF, S-CSCF, HSS	Используется для взаимодействия между I-CSCF/S-CSCF и HSS	Diameter
Dh	SIP AS, OSA, SCF, IM-SSF, HSS	Используется AS для поиска нужного HSS	Diameter
Dx	I-CSCF, S-CSCF, SLF	Используется I-CSCF/S-CSCF для поиска правильного HSS	Diameter
Gm	UE, P-CSCF	Используется для обмена сообщениями между UE и CSCF	SIP
Go	PDF, GGSN	Даёт возможность операторам управлять QoS на уровне передачи данных и обмениваться информацией между IMS и GPRS сетями	COPS (Rel5), Diameter (Rel6+)
Gq	P-CSCF, PDF	Используется для обмена политиками между P-CSCF и PDF	Diameter
ISC	S-CSCF, I-CSCF, AS	Используется для обмена сообщениями между CSCF и AS	SIP
Ma	I-CSCF -> AS	Используется для прямого перенаправления SIP-запросов, предназначенных серверам приложений (AS)	SIP
Mg	MGCF -> I-CSCF	MGCF преобразует сигнализацию ISUP в сигнализацию SIP и перенаправляет её в I-CSCF	SIP
Mi	S-CSCF -> BGCF	Используется для обмена сообщениями между S-CSCF и BGCF	SIP

Таблица 9.3

Описание стандартных интерфейсов (окончание)

Название интерфейса	Элементы IMS	Описание	Протокол
Mj	BGCF -> MGCF	Используется для обмена сообщениями между BGCF и MGCF в некоторых сетях IMS	SIP
Mk	BGCF -> BGCF	Используется для прямого обмена сообщениями между BGCFs и IMS	SIP
Mm	I-CSCF, S-CSCF, IP-сеть	Используется для обмена сообщениями между IMS и IP-сетями	-
Mn	MGCF, IM-MGW	Даёт возможность управлять ресурсами уровня передачи данных	H.248
Mp	MRFC, MRFP	Используется для обмена сообщениями между MRFC и MRFP	H.248
Mr	S-CSCF, MRFC	Используется для обмена сообщениями между S-CSCF и MRFC	SIP
Mw	P-CSCF, I-CSCF, S-CSCF	Используется для обмена сообщениями между несколькими CSCF	SIP
Rf	P-CSCF, I-CSCF, S-CSCF, BGCF, MRFC, MGCF, AS	Используется для offline-обмена информацией с CCF	Diameter
Ro	AS, MRFC	Используется для online-обмена информацией с ECF	Diameter
Sh	SIP AS, OSA SCS, HSS	Используется для обмена сообщениями между SIP AS/OSA SCS и HSS	Diameter
Si	IM-SSF, HSS	Используется для обмена сообщениями между IM-SSF и HSS	MAP
Sr	MRFC, AS	Используется MRFC для передачи документов (скриптов и др. ресурсов) для AS	HTTP
Ut	UE, AS (SIP AS, OSA SCS, IM-SSF)	Разрешает UE управлять информацией, касающейся его сервисов	HTTP(s)

9.5.2. Услуги в сетях IMS

Для реализации новых конвергентных услуг с гарантией качества обслуживания сервисная архитектура сети должна отвечать следующим требованиям:

- отделение уровней транспорта и доступа от сервисного уровня (прозрачность доступа);
- управление сеансом связи, в ходе которого задействуются несколько сервисов связи реального времени;
- совместимость с имеющимися сервисами интеллектуальной сети (IN), к которым относятся: определение имени вызывающей стороны, бесплатный номер (800), переносимость локального номера, сервисы, соответствующие стандартам CAMEL, ANSI-41 и т. д.;
- прозрачное взаимодействие с телефонными сетями (планы нумерации, сигнализация прохождения вызовов);
- конвергенция проводных и беспроводных сервисов;
- объединение голосовых услуг с сервисами реального времени (обмен мгновенными сообщениями);
- стандартизованные механизмы обмена пользовательской информацией между сервисами;

— стандартизованные механизмы аутентификации и биллинга конечных пользователей;
— стандартизованный, общий для всех сервисов графический пользовательский интерфейс;
— открытые стандартные интерфейсы и API для новых сервисов, разработанные сервис-провайдерами и третьими фирмами.

В сетях IMS определены следующие услуги:

— услуги, основанные на информации о присутствии и доступности пользователя — позволяют обеспечить доставку информации «правильному» человеку и/или на «правильное» устройство, т.е. при помощи протокола SIP можно обеспечить «прозрачное» переключение, например, между сотовой, WiFi или наземной связью с помощью одного устройства;
— услуги, основанные на информации о местоположении пользователя — позволяют предоставить пользователю информацию, актуальную для него в данный момент (например, прогноз погоды, информация о дорожной ситуации и т.п.);
— единый механизм авторизации, не связанный с конкретным устройством или технологией;
— управление групповыми списками;
— групповое общение (Group Communication);
— Push-To-Talk — услуга, работающая в полудуплексном (half-duplex) режиме, когда сотовый телефон используется как терминал системы профессиональной мобильной радиосвязи (основное преимущество — возможность «группового вызова», т.е. общения по принципу «один–многие»);
— Push-To-Show;
— доска для записей (Whiteboard) — услуга, позволяющая двум или нескольким абонентам совместно редактировать рисунки и документы в режиме реального времени. Все, что делается одним участником сеанса, видят в режиме on-line все остальные участники;
— многопользовательские игры в реальном времени (шахматы и другие игры);
— голосовые вызовы с усовершенствованными функциями (Enriched Voice Calling) — включают видеотелефонию и возможность добавления к вызовам своего контента;
— совместное использование файлов в сети (File Sharing);
— обеспечение необходимого уровня безопасности.

9.5.3. Протокол SIP

Протокол SIP предназначен для управления сеансами связи (инициация, модификация, завершение). Использование SIP в IMS позволяет реализовать услугу конференц-связи, поскольку любое число абонентов может динамически подключаться к сеансу и выходить из него. Кроме того, SIP даёт возможность динамически в рамках существующего сеанса связи подключать новые услуги (например, сеанс связи можно начать с текстового чата, потом добавить голосовую связь, а затем при необходимости и видео). Наконец, средства SIP способны при инициации или модификации сеанса связи учитывать характеристики канала доступа и терминала каждого пользователя и задействовать их оптимальным образом.

9.5.4. Преимущества и недостатки IMS

IMS обладает следующими преимуществами:

— предоставление множества услуг — нет жёсткой привязки средств управления услугами и способа их доставки до абонента с самими услугами, внедрение принципиально нового сервиса не требует построения соответствующей инфраструктуры для его доставки;
— хорошая масштабируемость сети оператора — модернизировать инфраструктуру сети можно поэлементно (например, при увеличении объёма трафика можно модернизировать только элементы уровня передачи данных, а при увеличении числа абонентов — элементы уровня управления);
— независимость IMS от специфики сетевого транспорта и каналов доступа делает её хорошей основой для конвергенции служб фиксированной и мобильной связи.

Недостатки IMS:

— для полноценного перехода к IMS операторам связи необходимо выстроить новую схему управления сеансами связи и модернизировать системы поддержки эксплуатации и бизнес-операций, а также обеспечить поддержку маршрутизаторами протокола IPv6;
— отсутствие терминалов, ориентированных на работу в IMS-сетях, — оконечное оборудование должно уметь инициировать и обрабатывать IMS-запросы, поддерживать работу сложных приложений;
— отсутствие поддержки non-SIP-приложений в рамках SIP-ориентированной архитектуры IMS.

9.6. Концепция A-IMS

В июле 2006 г. рабочая группа, в которую вошли ведущие поставщики телекоммуникационного оборудования Lucent Technologies, Cicso Systems, Motorola, Nortel и Qualcomm, под руководством оператора мобильной связи Verizon Wireless объявила о создании архитектуры *Advances to IMS (A-IMS)* [90, 91, 92].

Архитектура A-IMS [90, 92] является дальнейшим развитием стандарта IMS и призвана преодолеть его недостатки. Одной из проблем IMS является отсутствие поддержки non-SIP-приложений в рамках имеющейся SIP-ориентированной архитектуры IMS. Архитектура A-IMS позволяет осуществлять взаимодействие между SIP и non-SIP-приложениями, обеспечивает более полный policy-контроль над ними и управление сетевыми ресурсами, отвечающими за QoS, мобильность, безопасность, доступ и т.п. Включённые в архитектуру дополнения и усовершенствования применимы для построения сетей связи на основе разных технологий доступа (3G, xDSL WiMax, Cable) или конвергентных VoIP-сетей.

Основные элементы A-IMS:

— *подсистема управления приложениями (Application Manager, AM)* — элемент управления SIP-сессиями, выполняющий функции P-CSCF, I-CSCF, S-CSCF, BGCF;
— *подсистема управления данными об услугах (Services Data Manager, SDM)* — осуществляет хранение данных как для SIP, так и для non-SIP-приложений, включает в себя функциональность HSS и AAA, а также (опционально) SLF (Subscriber Location Function), KMF (Key Management Function) и Accounting;

— *подсистема управления несущей (Bearer Manager, BM)* — осуществляет контроль на уровне транспортного потока (несущей): контролирует применение соответствующих политик, правил, осуществляет управление потоками данных PFO (Packet Flow Optimization), идентификацию вторжений;
— *подсистема управления безопасностью (Security Manager, SM)* — выполняет задачи мониторинга событий в сети, обнаружения аномалий на основе программных алгоритмов, управления элементами сети для отражения угроз, управления IDS/IDP и политиками безопасности;
— *подсистема управления политиками (Policy Manager, PM)* — обеспечивает общее управление и контроль над распределением ресурсов сети (QoS, PFO, mobility, access и т.п.); поддерживает как SIP, так и non-SIP-приложения.

Дополнительные элементы A-IMS:

— *терминал доступа (Access Terminal, AT)* — оконечное устройство (фиксированное или мобильным), имеющее возможность предоставить доступ пользователей к услугам с помощью разных технологий (xDSL,WiFi, EVDO и т.п.);
— *шлюз IP (IP Gateway, IPGW)* — поддерживает взаимодействие между канальным и сетевым уровнями сети передачи данных, осуществляет аутентификацию устройств и переадресацию, отвечает за подсчёт пакетного трафика и обеспечение QoS;
— *посредник при предоставлении услуг (Service Broker, SB)* — представляет собой один из компонентов, отвечающих за механизм вызова (запуска) приложения с разных платформ (как использующих, так и не использующих SIP), при этом хранит логику предоставления услуг и управляет взаимодействием различных приложений на уровне сессий, являясь главным связующим звеном между SIP и non-SIP-приложениями;
— *функция управления ключами (Key Management Function, KMF)* — хранит ключи (абонентские и сетевые), которые используются при аутентификации абонентских устройств;
— *Regulatory and PSTN Servers* — обеспечивают интерфейс для выполнения определённых задач перехвата вызовов и сбора информации для компетентных ведомств.

9.7. Определение и суть NGN

Сеть связи следующего поколения (ССП — Next Generation Network, NGN) — концепция построения сетей связи, обеспечивающих предоставление неограниченного набора услуг с гибкими возможностями по их управлению, персонализации и созданию новых услуг за счёт унификации сетевых решений, предполагающая реализацию универсальной транспортной сети с распределённой коммутацией, вынесение функций предоставления услуг в оконечные сетевые узлы и интеграцию с традиционными сетями связи.

В сети NGN предоставляют широкий выбор технологий доступа, поставщиков услуг и самих услуг. Кроме того, в сетях NGN пользователи получают доступ к услугам независимо от местоположения и технического окружения, что позволяет обеспечить единообразие в предоставлении услуг.

9.7.1. Основополагающие характеристики NGN

Архитектура NGN предполагает чёткое разделение между функциями обслуживания и функциями транспортировки, что позволяет предоставлять и развивать как существующие, так и новые услуги вне зависимости от используемой сети и типа доступа.

Сети NGN обладают следующими основными характеристиками:

- пакетная коммутация;
- разделение ресурсов между пропускной способностью канала-носителя, вызовом/сеансом, приложением/услугами;
- разделение между предоставлением услуг и транспортировкой, предоставление открытых интерфейсов;
- поддержка широкого спектра услуг, приложений и механизмов на основе унифицированных блоков обслуживания (включая услуги в режиме реального масштабе времени, в потоковом или автономном режиме, мультимедийные услуги);
- возможности широкополосной передачи со сквозной функцией QoS;
- взаимодействие с существующими сетями посредством открытых интерфейсов;
- универсальная мобильность;
- неограниченный доступ пользователей к разным поставщикам услуг;
- разнообразие схем идентификации;
- единые характеристики обслуживания для одной и той же услуги с точки зрения пользователя;
- сближение услуг между фиксированной и подвижной связью;
- независимость связанных с обслуживанием функций от используемых технологий транспортировки;
- поддержка различных технологий «последней мили»;
- выполнение всех регламентных требований, например, для аварийной связи, защиты информации, конфиденциальности, законного перехвата и т.д.

9.7.2. Преимущества сетей, базирующихся на концепциях NGN

Сети NGN имеют ряд преимуществ (как для пользователей, так и для операторов связи) по сравнению с другими сетями:

- для оператора:
  - построение одной универсальной сети для оказания различных услуг;
  - возможность оптимального использования полосы пропускания для интеграции различных видов трафика и оказания различных услуг;
  - больше возможностей по расширению сети и спектра услуг;
  - простота в управлении и эксплуатации;
  - возможность быстрого внедрения новых услуг и приложений с различным требованием к объёму передаваемой информации и качеству её передачи;
- для пользователя:
  - абстрагирование от технологий реализации услуг электросвязи;
  - гибкое получение необходимого набора, объёма и качества услуг;
  - мобильность получения услуг.

9.7.3. Спектр предоставляемых услуг

В сетях NGN могут предоставляться следующие услуги:

- услуги службы телефонной связи:
  - местное телефонное соединение,
  - междугороднее телефонное соединение,
  - международное телефонное соединение,
  - передача факсимильных сообщений между терминальным оборудованием пользователей,
  - организация модемных соединений между терминальным оборудованием пользователей,
  - переадресация вызова,
  - индикация вызова,
  - удержание вызова;
- услуги служб передачи данных:
  - выделенный канал передачи данных,
  - постоянный и коммутируемый доступа в сеть Интернет,
  - виртуальные частные сети передачи данных;
- услуги телематических служб:
  - электронная почта,
  - голосовая почта,
  - доступ к информационным ресурсам,
  - телефония по IP-протоколу,
  - аудиоконференция и видеоконференция;
- услуги служб подвижной электросвязи;
- услуги поставщиков информации:
  - видео и аудио по запросу,
  - интерактивные новости,
  - электронный супермаркет,
  - дистанционное обучение и др.

9.7.4. Архитектура NGN

С функциональной точки зрения сеть следующего поколения делят на две плоскости — *плоскость услуг (Service Stratum)* и *транспортную плоскость (Transport Stratum)* (рис. 9.9) [93, 94].

Плоскость услуг включает функции, отвечающие за передачу *услугоориентированных данных (Service-Related Data)*, и функции, отвечающие за управление и эксплуатационную поддержку ресурсов услуг и услуг сети, необходимых для предоставления пользователю услуг и приложений.

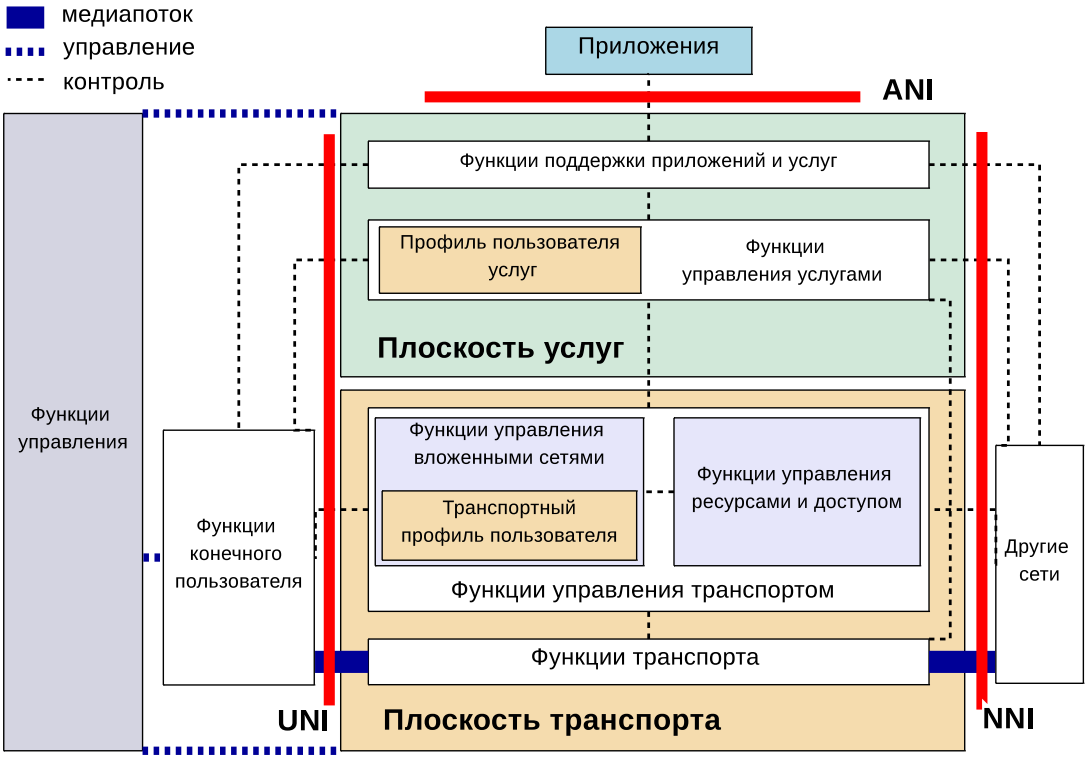


Рис. 9.9. Архитектура NGN

Транспортная плоскость включает функции, отвечающие за передачу данных, и функции, отвечающие за управление и эксплуатационную поддержку транспортных ресурсов для передачи этих данных между терминальными устройствами.

Взаимодействие приложений и элементов сети NGN осуществляется через *прикладной сетевой интерфейс (Application Network Interface, ANI)*.

Сетевой интерфейс пользователя (User Network Interface, UNI) обеспечивает взаимодействие функций конечного пользователя и элементов сети NGN.

Межсетевой интерфейс (Network Network Interface, NNI) обеспечивает взаимодействие сети NGN с другими сетями.

Так как сеть должна обеспечивать передачу разнородного трафика, в том числе чувствительного к задержкам, то немаловажными становятся такие требования к сети, как высокая надёжность оборудования узлов, поддержка функций управления трафиком, хорошая масштабируемость.

9.7.4.1. Функции транспортного уровня

Функции транспортного уровня обеспечивают соединение для всех компонентов и физически разделённых функций в рамках NGN, а также поддержку передачи медиаданных, контрольной и управляющей информации.

Определены следующие функции транспортного уровня:

— функции доступа к сети;
— функции граничного маршрутизатора;
— функции транзитного маршрутизатора;

— функции шлюза;
— функции обработки медианных;
— функции управления транспортным уровнем.

Функции доступа к сети (Access network functions) отвечают за доступ конечных пользователей в сеть, а также за сбор и оценку трафика, полученного транзитным узлом от пользователей. Кроме того, функции доступа к сети осуществляют управление качеством обслуживания, включая управление ёмкостью буфера, планирование и управление очередью, фильтрацию и классификацию трафика, его маркировку, применение политик по формированию трафика.

Функции доступа к сети классифицируются по технологиям доступа. Соответственно могут быть функции сетевого доступа по кабелю, оптоволокну, xDSL, по беспроводному соединению (IEEE 802.11 и 802.16 технологии, 3G RAN-доступ).

Функции граничного маршрутизатора (Edge functions) используются для обработки данных и трафика в случае, когда трафик, приходящий из различных сетей доступа, сливается в один поток на границе домена NGN. Сюда входят функции, связанные с поддержкой QoS и контролем трафика.

Функции транзитного маршрутизатора (Core transport functions) отвечают за передвижение информации через сеть NGN и предоставляют средства для разделения трафика относительно требований к качеству обслуживания. Данные функции предоставляют QoS-механизмы, непосредственно связанные с трафиком пользователя, включая управление буфером, размером очереди и планированием, фильтрацию пакетов, классификацию трафика, маркировку, разработку политик, контроль за точками доступа и возможностями брандмауэра.

Функции шлюза (Gateway functions) обеспечивают взаимодействие между функциями конечного пользователя и/или другими сетями, включая сети NGN, а также существующие сети, такие как, например, PSTN/ISDN, Интернет и т.д.

Функции обработки медианных (Media handling functions) предоставляют медиаресурсы, необходимые для предоставления услуг, таких как генерация тональных сигналов и преобразование одного кода в другой.

Функции управления транспортным уровнем (Transport control functions) включают в себя *функции контроля доступом к ресурсам (Resource and admission control functions, RACF)* и *функции контроля сетевых подключений (Network attachment control functions, NACFs)*.

Функции контроля доступа к ресурсам делают возможным представление для *функций управления услугами (Service Control Functions, SCF)* инфраструктуры транспортной сети в абстрактном виде и освобождают провайдеров от знания таких деталей, как топология сети, интерфейс подключения, потребление ресурсов, механизмы QoS. Данные функции осуществляют контроль за ресурсами сети на основе заданной политики, обеспечивают резервирование ресурсов, взаимодействуют с функциями маршрутизатора с целью контроля за выполнением функций по фильтрации пакетов, классификации трафика, маркировке, определению политики, управлению приоритетами и т.д.

Функции контроля сетевых подключений осуществляют регистрацию пользователя на уровне доступа и инициализацию функций пользователя, необходимых для доступа к услугам NGN. Кроме того, они идентифицируют транспортный уровень, управляют адресным пространством сети, аутентифицируют сессию доступа.

Таким образом, функции контроля сетевых подключений обеспечивают:
— динамическое предоставление IP-адресов и других параметров конфигурации;

— определение возможностей оборудования пользователя и других параметров;
— аутентификацию пользователя и сети на IP-уровне, а также взаимную аутентификацию пользователя и сети;
— авторизацию доступа в сеть на основе профиля пользователя;
— конфигурацию доступа в сеть на основе профиля пользователя.

9.7.4.2. Функции уровня управления услугами

Абстрактное представление функциональных групп на уровне управления услугами состоит:
— из функций управления услугами, включая функции профиля пользователя услуги;
— из функций поддержки приложений и функций поддержки услуг.

Функции управления услугами (Service control functions) включают в себя функции управления ресурсами, регистрацией, аутентификацией и авторизацией на уровне услуг. Также могут включать в себя функции управления медиаресурсами, т.е. специализированными ресурсами и шлюзами сигнализации.

Функции управления услугами размещают профили пользователя, представляющие собой информацию о пользователя и другую управляющую информацию, в единый профиль пользователя на уровне услуг в форме базы данных. Эти базы данных могут быть определены и реализованы как набор сообщающихся баз данных с функциональными средствами, расположенными в любой части NGN.

Функции поддержки приложений и функций поддержки услуг (Application support functions and service support functions) включают в себя функции маршрутизации, регистрации, аутентификации, авторизации на уровне приложений. Эти функции доступны как функциональный группе приложений, так и функциональной группе пользователей. Функции поддержки приложений и функции поддержки услуг работают совместно с функциями управления услугами для обеспечения пользователей и приложений теми NGN-услугами, которые им требуются.

9.7.4.3. Функции конечного пользователя

Интерфейсы пользователей и сетевые интерфейсы, соединённые с сетью доступа NGN, могут быть любыми. Оборудование пользователя может быть как фиксированным, так и мобильным.

9.7.4.4. Функции управления

Поддержка управления фундаментальна для работы в NGN. Эти функции дают возможность управлять NGN с целью обеспечения NGN услуг ожидаемого качества, безопасности и надёжности.

Функции управления распределены по всем *функциональным модулям (Functional Entity, FE)* и взаимодействуют с сетевыми элементами управления и элементами управления услугами.

Функции управления применяются как на транспортном уровне, так и на уровне услуг NGN. Для каждого уровня они затрагивают следующие области:
— управление исходными настройками;
— управление конфигурациями;
— управление учётными записями пользователей;

— управление производительностью;
— управление безопасностью.

Функции управления учётными записями пользователей дают возможность провайдеру обеспечивать пользователей заказанными ими услугами.

9.7.5. Концепции NGN

9.7.5.1. Уровень мобильности в архитектуре NGN

Архитектура NGN поддерживает возможность обеспечения мобильности пользователей внутри и между различными сетями доступа и сетями с технологией мобильного доступа. Мобильность может быть поддержана на различных уровнях архитектуры NGN.

9.7.5.2. Архитектура услуг NGN

Архитектура услуг NGN состоит из трёх различных функциональных областей: области приложений, области функций поддержки приложений и функций поддержки услуг на сервисном уровне, области ресурсов транспортного уровня NGN.

Область функций приложений может быть разбита на две категории — всё, что связано с сетевыми провайдерами, и иное. К первой группе относятся сетевые провайдеры, субпровайдеры и т.д. Ко второй — независимые провайдеры услуг, чей доступ к ресурсам должен быть аутентифицирован, контролируем и профильтрован функциями деблокиратора.

Посредством интерфейса ANI функциональная область функций поддержки приложений и услуг предлагает ресурсы услуг области приложений независимо от технологии сети. Также посредством ANI область приложений получает преимущества от использования возможностей и ресурсов функциональной области инфраструктуры NGN.

Архитектура услуг NGN следует трём основным функциональным характеристикам:

1) агностицизм — области функций поддержки приложений и функций поддержки услуг должны состоять из функций, независимых от инфраструктуры сети NGN;
2) поддержка официальных приложений и черт — архитектура услуг NGN не должна оказывать ограничивающее влияние на саму сеть NGN, т.е. должны поддерживаться функции по управлению сессиями, аутентификация, сведения о местонахождении и т.д.;
3) поддержка открытого интерфейса услуг — платформа услуг NGN должна предоставлять открытый интерфейс услуг (не зависящий от технологий транспортной сети), который обеспечивает доступ к таким функциям, как аутентификация, авторизация и безопасность, чтобы любой провайдер услуг мог воспользоваться возможностями сети.

9.7.5.3. Функции сокрытия сетевой топологии и просмотра трансляции сетевого адреса и порта

Сокрытие топологии уровня услуг достигается удалением или изменением топологической информации, передаваемой в прикладных сигнальных сообщениях

одноранговой сети (например, в SIP-основанных приложениях топологическая информация находится в SIP-заголовках).

Сокрытие топологии транспортного уровня достигается путём изменения топологической информации в пакетах данных или посредством блокировки сетевых контрольных пакетов с топологической информацией (например, изменение IP-адресов и/или номеров портов в пакетах данных, пересекающих границу между сетью доступа и доменом).

Просмотр трансляции сетевого адреса и порта (Network Address and Port Translation, NAPT) осуществляет просмотр удалённого NAPT в сетях доступа.

9.7.5.4. Контроль за переполнением

Для защиты функциональных модулей управления сессиями от концентрации нежелательных запросов необходима реализация на границе сетей доступа следующих функций: обнаружение концентрации запросов путём сбора информации от двух или нескольких функциональных модулей, передача полученной информации о концентрации запросов другим функциональным модулям, управление трафиком в соответствии с информацией о концентрации запросов.

9.7.5.5. Функции управления учётными записями пользователей и тарификацией

Функции управления учётными записями пользователей и тарификацией предназначены для представления обобщённой архитектуры предоставления провайдерами услуг пользователям. Они описывают условия сбора и обработки информации о пользователях и заказанных ими услугах для предоставления её NGN-провайдеру. Данные функции включают в себя *функцию сбора данных для тарификации (Charging Trigger Function, CTF)*, *функцию тарификации online (Online Charging Function, OCF)*, *функцию хранения информации для тарификации (Charging Collection Function, CCF)*, *функцию определения стоимости (Rating Function, RF)*, *функцию управления учётными записями пользователей (Account Management Function, AMF)*.

Функция сбора данных для тарификации осуществляет сбор данных о полученных пользователями ресурсах и услугах сети. Кроме того, данная функция создаёт учётные (тарификационные) события, используя онлайновый учёт. Данные направляются онлайновой учётной функции (OCF) для получения авторизации для доступа к ресурсам сети на основе запроса пользователя.

Функция хранения информации для тарификации получает сведения о произошедших событиях от CTF. Затем эта информация используется для формирования тарификационных данных, передаваемых биллинговым доменам.

Функция тарификации online получает данные от CTF и обрабатывают их практически в режиме реального времени для предоставления авторизации использования сетевых ресурсов на основе запроса пользователя. OCF предоставляет квоту на использование ресурсов, которая должна отслеживаться CTF.

Функция определения стоимости определяет стоимость предоставляемого сетевого ресурса.

Функция управления учётной записью хранит баланс учётной записи пользователя во время работы онлайновой учётной функции. Баланс учётной записи пользователя должен определять объём оставшегося доступного трафика, время или содержание, а также количество денег на счёте.

9.7.6. Компоненты сети NGN

На уровне услуг определены два компонента — компонент услуг IP-мультимедиа и компонент эмуляции сервиса PSTN/ISDN.

Компонент услуг IP-мультимедиа предоставляет посреднические услуги, включающие в себя управление услугами реального времени (голосовая или видео телефония, обмен сообщениями и т.п.), основанными на концепции IMS. В NGN IMS расширен для поддержки дополнительных видов сетей доступа, таких как xDSL и WLAN. Услуга имитации сетей PSTN/ISDN также обеспечивается этим компонентом.

Компонент эмуляции услуг PSTN/ISDN обеспечивает функционирование сетей на основе поддержки существующих услуг для интерфейсов пользователя и оборудования.

Эмуляция PSTN/ISDN относится к предоставлению услуг сетей PSTN/ISDN, используя адаптацию к IP-инфраструктуре. Компонент услуг эмуляции PSTN/ISDN делает возможным поддержку терминалов, связанных с IP-сетью, через шлюз. Все сервисы PSTN/ISDN остаются доступными и идентичными (т.е. с теми же операционными характеристиками), так что пользователи даже не подозревают, что они не соединены с TDM-основанным PSTN/ISDN.

На транспортном уровне также определены два компонента — *компонент функций контроля сетевых подключений (Network Attachment Control Functions, NACF)* и *компонент функций контроля доступом к ресурсам (Resource and Admission Control Functions, RACF)*.

Транспортные сети предоставляют соединения для всех компонентов и физически разделённых функций в рамках NGN. Транспортные сети подразделяются на сети доступа (Access Transport Networks) и внутренние сети (домены) (Core Transport Network) с шлюзом на границе, связывающим транспортные сети этих двух категорий. IP-соединение предоставляется оборудованию пользователя NGN на основе транспортных функций под контролем NACF- и RACF-компонент.

NGN взаимодействует с другими сетями, например, с PSTN/ISDN и Internet. Причём взаимодействие происходит как на уровне услуг, так и на транспортном уровне, посредством граничных шлюзов.

9.7.7. Softswitch и IMS как концепции NGN

Softswitch и IMS реализуют концепцию NGN. Однако следует отметить их принципиально разные подходы к реализации принципов NGN.

Softswitch ориентируется на жёсткую структуру построения сети — регламентируются структура сети, интерфейсы, все компоненты сети (протоколы, кодеки и пр.). При организации сети на базе Softswitch подлежат выполнению все требования стандарта.

IMS демонстрирует модульную архитектуру, регламентируя интерфейсы, оставляя свободу выбора в компонентах (протоколах, кодеков и пр.). Сеть на базе IMS можно реализовывать постепенно, добавляя новые элементы, что позволяет удешевить внедрение.

Подход Softswitch представляется более стройным, в то время как подход IMS — более гибким, что и вызвало ориентацию последних стандартов NGN именно на IMS.

Заключение

Достижения техники за последнее десятилетие привели к настоящему буму в области телекоммуникаций. Связь, находившаяся в статическом состоянии с середины 1980-х гг., сегодня превратилась в бурно развивающуюся отрасль. Сегодняшним клиентам рынка инфокоммуникационных услуг требуется широкий класс различных служб и приложений, предполагающий большое разнообразие протоколов, технологий и скоростей передачи. В существующей ситуации на рынке инфокоммуникационных услуг сети перегружены: они переполнены многочисленными интерфейсами клиентов, сетевыми слоями и контролируются слишком большим числом систем управления. При эволюции к прозрачной сети главной задачей является упрощение сети — это требование рынка и технологии.

На сегодняшний день развитие инфокоммуникационных услуг осуществляется в основном в рамках компьютерной сети Интернет, доступ к услугам которой происходит через традиционные сети связи. В то же время в ряде случаев услуги Интернета, ввиду ограниченных возможностей её транспортной инфраструктуры, не отвечают современным требованиям, предъявляемым к услугам информационного общества. В связи с этим развитие инфокоммуникационных услуг требует решения задач эффективного управления информационными ресурсами с одновременным расширением функциональности сетей связи. В свою очередь, это стимулирует процесс интеграции Интернета и сетей связи.

Список иллюстраций

Список таблиц

Используемая литература

1. *Hares S., Wittbrodt C.* An Echo Function for CLNP (ISO 8473), RFC 1575. — 1994. — http://www.faqs.org/rfcs/rfc1575.html.
2. *Никольский Н. Н.* Передача ОКС7 через IP // Сети и системы связи. — № 7. — 2005. — http://www.ccc.ru/magazine/depot/05_07/0301.htm.
3. *Самуйлов К. Е., Галентовская М.* Введение в систему сигнализации № 7 // Сети. — № 8-9. — 1999. — http://www.osp.ru/nets/1999/08-09/144246/_p1.html.
4. FCC, Part 68, Subpart F, Section 68.502. — http://www.fcc.gov/Bureaus/Engineering_Technology/Documents/cfr/1999/47cfr68.pdf.
5. *Лакнер Х.* Мобильность и полоса пропускания. Обзор актуальных стандартов IEEE 802 за последний год // LAN. — No 6. — 2007. — http://www.osp.ru/lan/2007/06/4238886/.
6. Bluetooth SIG, Inc. — http://www.bluetooth.com/Bluetooth/.
7. IEEE 802.15 Working Group for WPAN. — http://www.ieee802.org/15/.
8. Internet Protocol, RFC 791. — 1981. — http://www.faqs.org/rfcs/rfc791.html.
9. *Reynolds J., Postel J.* Assigned numbers, RFC 990. — 1986. — http://www.faqs.org/rfcs/rfc990.html.
10. *Reynolds J., Postel J.* Internet numbers, RFC 997. — 1987. — http://www.faqs.org/rfcs/rfc997.html.
11. *Braden R., Postel J.* Requirements for Internet gateways, RFC 1009. — 1987. — http://www.faqs.org/rfcs/rfc1009.html.
12. *Hinden R.* Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR), RFC 1517. — 1993. — http://www.faqs.org/rfcs/rfc1517.html.
13. *Rekhter Y., Li T.* An Architecture for IP Address Allocation with CIDR, RFC 1518. — 1993. — http://www.faqs.org/rfcs/rfc1518.html.
14. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, RFC 1519 / V. Fuller, T. Li, J. Yu, K. Varadhan. — 1993. — http://www.faqs.org/rfcs/rfc1519.html.
15. *Rekhter Y., Topolcic C.* Exchanging Routing Information Across Provider Boundaries in the CIDR Environment, RFC 1520. — 1993. — http://www.faqs.org/rfcs/rfc1520.html.
16. *Deering S., Hinden R.* Internet Protocol, Version 6 (IPv6) Specification, RFC 2460. — 1998. — http://www.faqs.org/rfcs/rfc2460.html.
17. *Deering S., Hinden R.* Internet Protocol, Version 6 (IPv6) Specification, RFC 1883. — 1995. — http://www.faqs.org/rfcs/rfc1883.html.
18. *Hinden R., Deering S.* Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3513. — 2003. — http://www.faqs.org/rfcs/rfc3513.html.
19. *Blanchet M.* A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block, RFC 3531. — 2003. — http://www.faqs.org/rfcs/rfc3531.html.
20. *Hinden R., Deering S., Nordmark E.* IPv6 Global Unicast Address Format, RFC 3587. — 2003. — http://www.faqs.org/rfcs/rfc3587.html.
21. OSI NSAPs and IPv6, RFC 1888 / J. Bound, B. Carpenter, D. Harrington et al. — 1996. — http://www.faqs.org/rfcs/rfc1888.html.

22. *Narten T., Draves R.* Privacy Extensions for Stateless Address Autoconfiguration in IPv6, RFC 3041. — 2001. — http://www.faqs.org/rfcs/rfc3041.html.
23. *Postel J.* Internet Control Message Protocol, RFC 792. — 1981. — http://www.faqs.org/rfcs/rfc792.html.
24. *Plummer D. C.* Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, RFC 826. — 1982. — http://www.faqs.org/rfcs/rfc826.html.
25. A Reverse Address Resolution Protocol, RFC 903 / R. Finlayson, T. Mann, J. Mogul, M. Theimer. — 1984. — http://www.faqs.org/rfcs/rfc903.html.
26. The Click Modular Router Project. — http://www.read.cs.ucla.edu/click/.
27. *Rosen E., Viswanathan A., Callon R.* Multiprotocol Label Switching Architecture, RFC 3031. — 2001. — http://www.ietf.org/rfc/rfc3031.txt.
28. MPLS Label Stack Encoding, RFC 3032 / E. Rosen, D. Tappan, G. Fedorkow et al. — 2001. — http://www.ietf.org/rfc/rfc3032.txt.
29. LDP Specification, RFC 3036 / L. Andersson, P. Doolan, N. Feldman et al. — 2001. — http://www.ietf.org/rfc/rfc3036.txt.
30. *Postel J.* User Datagram Protocol, RFC 768. — 1980. — http://www.faqs.org/rfcs/rfc768.html.
31. *Postel J.* Transmission Control Protocol, RFC 793. — 1981. — http://www.faqs.org/rfcs/rfc793.html.
32. Stream Control Transmission Protocol, RFC 2960 / R. Stewart, Q. Xie, K. Morneault et al. — 2000. — http://www.faqs.org/rfcs/rfc2960.html.
33. *Ong L., Yoakum J.* An Introduction to the Stream Control Transmission Protocol (SCTP), RFC 3286. — 2002. — http://www.faqs.org/rfcs/rfc3286.html.
34. *Mogul J., Deering S.* Path MTU discovery, RFC 1191. — 1990. — http://www.faqs.org/rfcs/rfc1191.html.
35. *Floyd S., Handley M., Kohler E.* Problem Statement for the Datagram Congestion Control Protocol (DCCP), RFC 4336. — 2006. — http://tools.ietf.org/html/rfc4336.
36. *Kohler E., Handley M., Floyd S.* Datagram Congestion Control Protocol (DCCP), RFC 4340. — 2006. — http://tools.ietf.org/html/rfc4340.
37. *Ramakrishnan K., Floyd S., Black D.* The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168. — 2001. — http://tools.ietf.org/html/rfc3168.
38. *Mockapetris P.* Domain names: Concepts and facilities, RFC 882. — 1983. — http://www.faqs.org/rfcs/rfc882.html.
39. *Mockapetris P.* Domain names: Implementation specification, RFC 883. — 1983. — http://www.faqs.org/rfcs/rfc883.html.
40. *Mockapetris P.* Domain names: Concepts and facilities, RFC 1034. — 1987. — http://www.faqs.org/rfcs/rfc1034.html.
41. *Mockapetris P.* Domain names: Implementation specification, RFC 1035. — 1987. — http://www.faqs.org/rfcs/rfc1035.html.
42. *Ohta M.* Incremental Zone Transfer in DNS, RFC 1995. — 1996. — http://www.faqs.org/rfcs/rfc1995.html.
43. *Vixie P.* A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY), RFC 1996. — 1996. — http://www.faqs.org/rfcs/rfc1996.html.

44. Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136 / P. Vixie, S. Thomson, Y. Rekhter, J. Bound. — 1997. — http://www.faqs.org/rfcs/rfc2136.html.
45. *Droms R.* Dynamic Host Configuration Protocol, RFC 1541. — 1993. — http://www.faqs.org/rfcs/rfc1541.html.
46. *Ефимушкин В., Ледовских Т.* От E.164 к ENUM // Электросвязь. — № 07. — 2002. — С. 9–14.
47. *Faltstrom P.* E.164 number and DNS, RFC 2916. — 2000. — http://www.faqs.org/rfcs/rfc2916.html.
48. RFC2916 E.164 Number and DNS. — http://www.ietf.org/rfc/rfc2916.txt.
49. *Гринфилд Д.* За кулисами рынка IP-телефонии // LAN. — № 04. — 2002. — http://www.osp.ru/lan/2002/04/136017/.
50. Low Latency Queueing. — 2001. — http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt2/qcfconmg.htm#1001280.
51. *Floyd S., Jacobson V.* Random Early Detection gateways for Congestion Avoidance // IEEE/ACM Transactions on Networking. — Vol. 1, No 4. — 1993. — Pp. 397–413. — http://www.icir.org/floyd/papers/red/red.html.
52. *Floyd S., Gummadi R., Shenker S.* Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management. — 2001. — http://www.icir.org/floyd/papers/adaptiveRed.pdf.
53. *Braden R., Clark D., Shenker S.* Integrated Services in the Internet Architecture: an Overview, RFC 1633. — 1994. — http://www.faqs.org/rfcs/rfc1633.html.
54. *Wroclawski J.* The Use of RSVP with IETF Integrated Services, RFC 2210. — 1997. — http://www.faqs.org/rfcs/rfc2210.html.
55. *Shenker S., Partridge C., Guerin R.* Specification of Guaranteed Quality of Service, RFC 2212. — 1997. — http://www.faqs.org/rfcs/rfc2212.html.
56. *Wroclawski J.* Specification of the Controlled-Load Network Element Service, RFC 2211. — 1997. — http://www.faqs.org/rfcs/rfc2211.html.
57. Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification, RFC 2205 / R. Braden, L. Zhang, S. Berson et al. — 1997. — http://www.faqs.org/rfcs/rfc2205.html.
58. *Herzog S.* RSVP Extensions for Policy Control, RFC 2750. — 2000. — http://www.faqs.org/rfcs/rfc2750.html.
59. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474 / K. Nichols, S. Blake, F. Baker, D. Black. — 1998. — http://tools.ietf.org/html/rfc2474.txt.
60. An Architecture for Differentiated Services, RFC 2475 / S. Blake, D. Black, M. Carlson et al. — 1998. — http://tools.ietf.org/html/rfc2475.txt.
61. Assured Forwarding PHB Group, RFC 2597 / J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. — 1999. — http://tools.ietf.org/html/rfc2597.
62. *Jacobson V., Nichols K., Poduri K.* An Expedited Forwarding PHB, RFC 2598. — 1999. — http://tools.ietf.org/html/rfc2598.
63. ITU-T Recommendation H.323 v1: Packet-based multimedia communications systems. — 1996.
64. ITU-T Recommendation Q.931: ISDN user-network interface layer 3 specification for basic call control. — 1998.
65. ITU-T Recommendation H.245, Control protocol for multimedia communication. — 2006.

66. ITU-T Recommendation G.711 Pulse code modulation (PCM) of voice frequencies. — 1988.
67. ITU-T Recommendation G.723.1 Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. — 1996, 2006.
68. Federal Standard 1016, Telecommunications: Analog to Digital Conversion of Radio Voice by 4,800 bit/second Code Excited Linear Prediction (CELP). — Washington: National Communications System, Office of Technology and Standards, 1991.
69. NCS Technical Information Bulletin 92-1. Details to Assist in Implementation of Federal Standard 1016 CELP.
70. ITU-T Recomendation G.728 Coding of speech at 16 kbit/s using law delay code excited linear prediction. — 1992.
71. ITU-T Recommendation G.729 Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction. — 1996.
72. ITU-T Recommendation G.729 — Annex A, Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACLEP), Annex A: Reduced complexity 8 kbit/s CS-ACELP speech codec. — 1996.
73. ITU-T Recommendation G.729 — Annex B. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACLEP), Annex B: A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70. — 1996.
74. ITU-T Recommendation H.261: Video CODEC for audiovisual services at p X 64 kbit/s. — 1993.
75. Discrete-time signal processing. — 2nd edition upper saddle river edition. — NJ, USA: Prentice-Hall, Inc., 1999.
76. Алгоритмические основы растровой графики / Д. В. Иванов, А. С. Карпов, Е. П. Кузьмин и др. — Изд. ИНТУИТ, 2007. — 286 с.
77. ITU-T Recommendation H.263: Video coding for low bit rate communication. — 2005.
78. ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services. — 2005.
79. ISO/IEC 14496-10 Standart «Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding». — 2005.
80. ITU-T Recommendation T.120: Transmission protocols for multimedia data. — 1996.
81. *Иванцов И.* Стеки протоколов // Журнал сетевых решений LAN. — № 3. — 2007.
82. ITU-T Recommendation H.225.0, Call signalling protocols and media stream packetization for packet-based multimedia communication systems. — 2006.
83. ITU-T Recommendation H.235.0, H.323 Security: Framework for security in H-series (H.323 and other H.245-based) multimedia systems. — 2005.
84. ITU-T Recommendation H.450.2: Call transfer supplementary service for H.323. — 1998.
85. ITU-T Recommendation H.450.3: Call diversion supplementary service for H.323. — 1998.
86. SIP: Session Initiation Protocol, RFC 2543 / M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. — 1999.
87. SIP: Session Initiation Protocol, RFC 3261 / J. Rosenberg, H. Schulzrinne, G. Camarillo et al. — 2002.
88. *Handley M., Jacobson V.* SDP: Session Description Protocol, RFC 2327. — 1998.

89. *Гулевич Д. С.* Сети связи следующего поколения. — Открытые системы, ИНТУИТ, 2007. — http://www.intuit.ru/department/network/ndnets/.
90. *Duffy J.* Verizon Wireless leads group offering IMS extensions // NetworkWorld.com. — 2006. — http://www.networkworld.com/news/2006/072706-verizon-wireless-ims.html.
91. *Duffy J.* Verizon, others offer IMS extensions: Now comes the hard part for A-IMS // NetworkWorld.com. — 2006. — http://www.networkworld.com/news/2006/073106-a-ims.html.
92. *Хисматуллин И.* IMS предлагается дополнить // Сети. — № 14. — 2006. — http://www.osp.ru/nets/2006/14/3199456/.
93. ITU-T Recommendation Y.2011: General principles and general reference model for Next Generation Networks. — 2004. — http://www.itu.int/rec/T-REC-Y.2011-200410-I/en.
94. ITU-T Recommendation Y.2012: Functional requirements and architecture of the NGN, release 1. — 2006. — http://www.itu.int/rec/T-REC-Y.2012-200609-I/en.
95. В Интернет через Ethernet. От соединения двух компьютеров до сети микрорайона // NAG.ru. Все об Ethernet провайдинге. — — http://www.nag.ru/goodies/book/.
96. *Воловодов А.* От тактовой частоты до информационной магистрали // Сети и системы связи. — № 9. — 1999. — http://www.ccc.ru/magazine/depot/99_09/read.html?0101.htm.
97. Беспроводные сети Wi-Fi / А. В. Пролетарский, И. В. Баскаков, Р. А. Федотов и др. — Изд-во «Интернет-университет информационных технологий — ИНТУИТ.ру», БИНОМ, 2007. — 216 с. — http://www.intuit.ru/department/network/wifi/.
98. Сети Fast Ethernet // Оптилинк. — 2001. — http://www.optilink.ru/Techdoc/Ethernet/fast_ethernet.htm.
99. *Смелянский Р. Л.* Системы передачи данных и сети ЭВМ. — — http://www.kgtu.runnet.ru/WD/TUTOR/cn/index.html.
100. Основы локальных сетей. — 2005. — http://www.intuit.ru/department/network/baslocnet/.
101. *Кунегин С. В.* Общее описание метода информационного обмена в сетях передачи данных Frame Relay. — 1998. — http://kunegin.narod.ru/ref/fro/index.htm.
102. *Филимонов А.* Сети Frame Relay // Сети ЭВМ и телекоммуникации (курс лекций). — — http://lectures.net.ru/lectures/.
103. *Мельников Д.* Frame relay для профессионалов и не только // Сети. — № 10. — 1997. — http://www.osp.ru/nets/1997/10/142934/.
104. *Андронов С.* О структуре и свойствах современных пакетных сетей // JetInfo. — № 6(73). — 1999. — http://www.jetinfo.ru/1999/6/1/article1.6.1999.html.
105. *Афонцев Э.* Metro Ethernet. Архитектура и технологии // NAG.ru. — 2005. — http://www.nag.ru/2005/0227/0227.shtml.
106. *Афонцев Э.* Назад в будущее. Metro Ethernet // NAG.ru. — 2005. — http://www.nag.ru/2005/0212/0212.shtml.
107. Bluetooth в целом // Rainbow Technologies. — 2005. — http://www.rtcs.ru/article_detail.asp?id=331.
108. *Широков Ф.* Bluetooth: на пути к миру без проводов // Открытые системы. — № 2. — 2001. — http://www.radioscanner.ru/info/article95/.

109. *Невдяев Л.* Bluetooth — королевская технология // Сети. — № 10. — 2000. — http://www.osp.ru/nets/2000/10/141423/.
110. *Митилино С.* Беспроводные сети Bluetooth // Интернет и сети. — 2002. — http://itc.ua/node/11177/.
111. *Кессених В., Иванов Е., Кондрашов З.* Bluetooth: принципы построения и функционирования // Chip NEWS. — 2001. — http://www.chip-news.ru/archive/chipnews/200107/8.html.
112. *Редькин А.* Замена Bluetooth // Сети и Телекоммуникации. — 2005. — http://www.citforum.ncstu.ru/nets/wireless/wireless_usb/.
113. Краткий обзор протоколов информационно-вычислительных сетей. — 1999. — http://cdo.bseu.by/library/ibs1/net_1/tcp_ip/net/frmp_ipv6.htm.
114. *Юшков Т.* Архитектура MPLS. — 2005. — http://www.mpls-exp.ru/mplsarchitecture.html.
115. *Захватов М.* Качество обслуживания в операторских сетях. — http://www.opennet.ru/docs/RUS/qos_oper/.
116. A Single Rate Three Color Marker, RFC 2697. — 1999. — http://www.ietf.org/rfc/rfc2697.txt.
117. A Two Rate Three Color Marker, RFC 2698. — 1999. — http://www.ietf.org/rfc/rfc2698.txt.
118. A Time Sliding Window Three Color Marker (TSWTCM), RFC 2859. — 2000. — http://www.ietf.org/rfc/rfc2859.txt.
119. *Grossman D.* New Terminology and Clarifications for Diffserv, RFC 3260. — 2002. — http://tools.ietf.org/html/rfc3260.
120. *Almquist P.* Type of Service in the Internet Protocol Suite, RFC 1349. — 1992. — http://tools.ietf.org/html/rfc1349.
121. *Brim S., Carpenter B., Faucheur F. L.* Per Hop Behavior Identification Codes, RFC 2836. — 2000. — http://tools.ietf.org/html/rfc2836.
122. Per Hop Behavior Identification Codes, RFC 3140 / D. Black, S. Brim, B. Carpenter, F. L. Faucheur. — 2001. — http://tools.ietf.org/html/rfc3140.
123. ITU-T Recommendation H.450.1, Generic functional protocol for the support of supplementary services in H.323. — 1998.
124. ITU-T Recommendation H.450.4: Call hold supplementary service for H.323. — 1999.
125. ITU-T Recommendation H.450.5: Call park and call pickup supplementary services for H.323. — 1999.
126. ITU-T Recommendation H.450.6: Call waiting supplementary service for H.323. — 1999.
127. ITU-T Recommendation H.450.7: Message waiting indication supplementary service for H.323. — 1999.
128. ITU-T Recommendation H.450.8: Name identification supplementary service for H.323. — 2000.
129. *Вегешна Ш.* Качество обслуживания в сетях IP. — 2003. — 368 с.
130. 3GPP TS 23.228. IP Multimedia Subsystem (IMS): Stage 2 (Release 8). — 2007.
131. 3GPP TS 23.517. Protocols for Advanced Networking (TISPAN); IP Multimedia Subsystem (IMS); Functional architecture (Release 8). — 2007.
132. *Таненбаум Э.* Компьютерные сети. — Четвёртое издание. — Питер: Питер, 2007. — С. 992.
133. *Семёнов А. Ю.* Протоколы Интернет. Энциклопедия 2-е изд. — 2005. — С. 1100.

134. *Семёнов А. Ю.* Алгоритмы телекоммуникационных сетей. — Изд-во Интернет-университет информационных технологий, Бином, 2007.
135. *Гольдштейн А. Б., Гольдштейн Б. С.* Softswitch. — БХВ — Санкт-Петербург, 2006.
136. *Самуйлов К. Е.* Методы анализа и расчёта сетей ОКС 7: Монография. — М.: Изд-во РУДН, 2002.
137. *Халсалл Ф.* Передача данных, сети компьютеров и взаимосвязь открытых систем. — М.: Радио и связь, 1995. — С. 408.
138. *Олифер В. Г., Олифер Н. А.* Компьютерные сети. Принципы, технологии, протоколы. Учебник для ВУЗов 3-е издание. — Питер: Питер, 2007. — С. 960.
139. *Олифер В. Г., Олифер Н. А.* Основы сетей передачи данных. Курс лекций. — Издательство: Интернет-университет информационных технологий, Бином, 2005. — С. 176.
140. *Хогдал Д. С.* Анализ и диагностика компьютерных сетей. — М.: Издательство «Лори», 2001. — С. 354.
141. *Кульгин М.* Практика построения компьютерных сетей. Для профессионалов. — СПб.: Питер, 2001. — С. 320.
142. *Кульгин М.* Технология корпоративных сетей. Энциклопедия. — СПб.: Питер, 2000. — С. 704.
143. *Уолренд Д.* Телекоммуникационные и компьютерные сети. Вводный курс. — М.: Постмаркет, 2001. — С. 480.
144. *Камер Д. Э.* Компьютерные сети и Internet. Разработка приложений для Internet. — Третье издание. — М.: Издательский дом «Вильямс», 2002. — С. 640.
145. *Снейдер Й.* Эффективное программирование TCP/IP. Библиотека программиста. — СПб.: Питер, 2001. — С. 320.
146. Рекомендация МСЭ-Т E.164 Международной план нумерации электросвязи общего пользования. — 2005. — `http://www.itu.int/rec/T-REC-E.164-200502-I/ru`.
147. *Федорушкин И.* ENUM – нужна ли России альтернативная нумерация? // Intelligent Enterprise. — № 03. — 2006. — `http://www.rtcomm.ru/about/press/pub/926.html`.

Рекомендуемая литература

1. В Интернет через Ethernet. От соединения двух компьютеров до сети микрорайона. — http://www.nag.ru/goodies/book/.
2. *Вегешна Ш.* Качество обслуживания в сетях IP. — 2003. — 368 с.
3. *Гольдштейн А. Б., Гольдштейн Б. С.* Softswitch. — БХВ — Санкт-Петербург, 2006.
4. *Гулевич Д. С.* Сети связи следующего поколения. — Открытые системы, ИНТУИТ, 2007. — http://www.intuit.ru/department/network/ndnets/.
5. *Камер Д. Э.* Компьютерные сети и Internet. Разработка приложений для Internet. — Третье издание. — М.: Издательский дом «Вильямс», 2002. — С. 640.
6. *Кульгин М.* Технология корпоративных сетей. Энциклопедия. — СПб.: Питер, 2000. — С. 704.
7. *Кульгин М.* Практика построения компьютерных сетей. Для профессионалов. — СПб.: Питер, 2001. — С. 320.
8. *Олифер В. Г., Олифер Н. А.* Основы сетей передачи данных. Курс лекций. — Издательство: Интернет-университет информационных технологий, Бином, 2005. — С. 176.
9. *Олифер В. Г., Олифер Н. А.* Компьютерные сети. Принципы, технологии, протоколы. Учебник для ВУЗов 3-е издание. — Питер: Питер, 2007. — С. 960.
10. *Самуйлов К. Е.* Методы анализа и расчёта сетей ОКС 7: Монография. — М.: Изд-во РУДН, 2002.
11. *Семёнов А. Ю.* Протоколы Интернет. Энциклопедия 2-е изд. — 2005. — С. 1100.
12. *Семёнов А. Ю.* Алгоритмы телекоммуникационных сетей. — Изд-во Интернет-университет информационных технологий, Бином, 2007.
13. *Снейдер Й.* Эффективное программирование TCP/IP. Библиотека программиста. — СПб.: Питер, 2001. — С. 320.
14. *Таненбаум Э.* Компьютерные сети. — Четвёртое издание. — Питер: Питер, 2007. — С. 992.
15. *Уолренд Д.* Телекоммуникационные и компьютерные сети. Вводный курс. — М.: Постмаркет, 2001. — С. 480.
16. *Халсалл Ф.* Передача данных, сети компьютеров и взаимосвязь открытых систем. — М.: Радио и связь, 1995. — С. 408.
17. *Хогдал Д. С.* Анализ и диагностика компьютерных сетей. — М.: Издательство «Лори», 2001. — С. 354.

Предметный указатель

ОПИСАНИЕ КУРСА И ПРОГРАММА

1. Цели и задачи курса

Область знаний

Курс относится к области знаний «Информационно-телекоммуникационные системы», соответствующей одноименному приоритетному направлению развития науки и технологий, входящему в перечень, утвержденный Президентом Российской федерации.

Уровень обучения и направления подготовки по действующему перечню

Курс относится к программе дополнительной профессиональной подготовки для студентов направлений 550200 «Автоматизация и управление», 511200 «Математика, прикладная математика», 510400 «Физика», 521500 «Менеджмент», 521600 «Экономика» 060800 «Экономика и управление на предприятии (по отраслям производства)».

Курс является составляющей модуля программы дополнительной профессиональной подготовки «Основы управления инфокоммуникациями», которая включает также курсы:

«Введение в формальные методы описания бизнес-процессов»;

«Введение в управление инфокоммуникациями»;

«Корпоративные информационные системы».

Учащиеся, успешно освоившие данную программу дополнительной профессиональной подготовки, могут поступать на магистерскую программу «Управление инфокоммуникациями».

Цели курса

- Ввести учащихся в предметную область существующих систем и сетей телекоммуникаций.
- Сформировать понятийный аппарат в области концепций, архитектур, стандартов современных систем и сетей телекоммуникаций.
- Ознакомить слушателей с новыми технологиями в области систем и сетей телекоммуникаций.
- Создать у слушателей понимание принципов построения современных систем и сетей телекоммуникаций.

Задачи курса

После успешного прохождения курса слушатели должны

знать:

- базовые понятия систем и сетей телекоммуникаций;
- общие характеристики концепций, архитектур, стандартов современных систем и сетей телекоммуникаций;
- основные принципы управления современными системами и сетями телекоммуникаций;

уметь:

- квалифицированно и грамотно оперировать базовыми терминами и понятиями;
- представить свои знания в формализованном виде;
- использовать полученные знания при общей характеристике современного состояния систем и сетей телекоммуникаций.

2. Инновационность курса

По содержанию.

Современные методы проектирования сетей и систем телекоммуникаций в частности базируются на новейших достижениях целого ряда научных областей, обеспечивающих развитие приоритетного направления развития науки и технологий – информационно-телекоммуникационные технологии, – входящего в перечень, утвержденного Президентом Российской федерации. К этим областям в первую очередь относятся информационная интеграция, информационно-телекоммуникационные системы и искусственный интеллект. Последние достижения в этой области сконцентрированы в целом ряде концепций, архитектурных моделей и методологий, принятых на международном уровне в виде стандартов и рекомендаций, разработанных ведущими производителями и исследовательскими центрами. Эти концепции в свою очередь опираются на другие новейшие достижения в области инфокоммуникационных технологий.

Содержание курса обеспечивает слушателей необходимым объёмом знаний для освоения основ построения и эксплуатации современных сетей и систем телекоммуникаций.

По методике преподавания и организации учебного процесса.

Методика преподавания основана на применении современных информационных технологий. Учебно-методический комплекс с одноименным названием помимо традиционных методических материалов включает электронный учебник, интегрированный в инфокоммуникационную среду типа eLearning. Эти средства позволяют организовать и провести лабораторные занятия в виде виртуального класса, где студенты работают под руководством преподавателя в асинхронном режиме. Такой режим позволяет осуществлять эффективный контроль уровня знаний за счет постоянного

наблюдения за степенью освоения курса учащимися и за ходом выполнения промежуточных видов контроля знаний.

По литературе.

В настоящее время основная масса литературных источников описывает предметную область либо слишком абстрактно, либо углубляясь в несущественные детали. Учебная литература на русском языке по большей части потеряла актуальность и содержит устаревшие данные. Современное состояние предметной области доступно в основном в литературе на иностранных языках (английском).

3. Структура курса

Трудоемкость курса: 4 кредита.

Аудиторные занятия:

лекции – 2 часа в неделю;

лабораторные занятия – 2 часа в неделю;

Самостоятельная работа студента: 1 час в неделю.

Содержание курса, объём знаний, общие требования к промежуточному и итоговому контролю знаний определяются программой курса, график обучения определяется календарным планом, а оценка освоения программы курса студентом – методикой оценки уровня знаний.

Программа курса

Темы лекций

Тема 1. Общая характеристика проблемной области. Базовые понятия в области систем и сетей телекоммуникаций. Стандартизирующие организации.

В данной теме даются и объясняются такие базовые понятия систем телекоммуникаций, как протокол, интерфейс, служба. Дается обзор существующих сетей связи, сетевых сервисов. Рассматривается структура и основные аспекты деятельности стандартизирующих организаций, таких как ISO (International Organization for Standardization — Международная организация по стандартизации), ITU (International Telecommunication Union — Международный телекоммуникационный союз), IEEE (Institute of Electrical and Electronic Engineers — Институт инженеров по электротехнике и радиоэлектронике) и др.

Тема 2. Модель ISO/OSI. Иерархия протоколов различных стеков относительно модели ISO/OSI.

В данной теме рассматриваются общие принципы построения модели взаимодействия открытых систем (ISO/OSI), иерархия протоколов различных стеков протоколов (TCP/IP, IEEE, ISO/OSI, H.323, SS7 и др.) по отношению к модели ISO/OSI.

Тема 3. Физический уровень модели ISO/OSI.

В данной теме рассматриваются методы и технологии физического уровня модели ISO/OSI. В частности, дается обзор возможных сред передачи (в том числе и стандарты кабельной системы), методов кодирования сигнала и сферы их применения.

Тема 4. Канальный уровень модели ISO/OSI.

В данной теме рассматриваются протоколы, методы и технологии канального уровня модели ISO/OSI. В частности, изучаются методы и протоколы (семейства ALOHA и CSMA) доступа к среде, а также технологии сетей (Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, 100VG–AnyLAN, Wireless Networks, WiMAX, Bluetooth, FDDI, Frame Relay). Упор делается на стандарты IEEE 802.x.

Тема 5. Сетевой уровень модели ISO/OSI.

В данной теме рассматриваются протоколы межсетевого уровня стека протоколов TCP/IP. Особое внимание уделяется протоколу IP: изучается формат кадра IP, IP-адресация (IPv4 и IPv6), взаимодействие межсетевого уровня с физическим. Далее отдельным пунктом изучается проблема маршрутизации: классификация алгоритмов маршрутизации и, собственно, протоколы статической (iproute2, click) и динамической маршрутизации (RIP, OSPF, BGP), сфера их применения, достоинства и недостатки. Кратко рассматриваются другие протоколы межсетевого уровня стека протоколов TCP/IP (ARP, RARP, ICMP), их назначение.

Тема 6. Транспортный уровень модели ISO/OSI.

В данной теме рассматриваются протоколы транспортного уровня стека протоколов TCP/IP: TCP (формат TCP-пакета, алгоритм установления связи (сессия TCP), параметры передачи, MTU, надежная доставка, технология скользящего окна, процедура «медленный старт»), UDP (формат UDP-пакета, псевдозаголовок, ненадежная передача), DCCP, SCTP.

Тема 7. Протоколы верхних уровней модели ISO/OSI.

В данной теме рассматривается служба доменных имен (DNS), а также расширение ENUM для поддержки E.164.

Тема 8. QoS и передача мультимедийных данных.

С расширением услуг, предоставляемых сетями передачи данных, и изменением характера передаваемых данных (уменьшение доли трафика чисто файловых протоколов, таких как FTP, в пользу протоколов передачи мультимедийных данных) критичным становится обеспечение качества обслуживания (QoS) передачи. В данной теме вводятся базовые понятия QoS – доступность, потери, задержка, пропускная способность и их значение для различных типов трафика. Рассматриваются специальные решения

обеспечения QoS, такие как организация виртуальных каналов в ATM, а также решения для IP-сетей и Ethernet – организация виртуальных каналов при помощи меток (MPLS), разбиение трафика на классы в соответствии с приоритетами каждого типа трафика и определение политик обслуживания этих классов трафика (DiffServ и IntServ). В связи с этим рассматриваются инструменты классификации и маркировки пакетов (CoS, ToS, DSCP, MPLS EXP), а также механизмы планирования (WFQ, RED, LLQ и др.) и механизмы канального уровня (policing – ограничение скорости и shaping – выравнивание трафика, фрагментация и чередование пакетов, компрессия).

Тема 9. Мультисервисные сети.

В данной теме изучаются некоторые мультисервисные сети (ISDN, H.323), их переход в конвергентные сети. В исторической ретроспективе рассматриваются два основных подхода к построению конвергентных сетей, называемых также сетями следующего поколения (Next Generation Networks – NGN), – Softswitch и IMS. Дается сравнительный обзор концепций Softswitch и IMS как развитие концепций конвергирования сетей коммутации каналов (в частности, ТфОП) и сетей коммутации пакетов (в частности, IP-сети). Описываются архитектурный особенности и основные протоколы обоих подходов.

Данная тема призвана представить общую концепцию технологий построения сетей NGN. Выделяются основные группы протоколов: протоколы кодирования (G.711, G.723.1, G.728, G.729, H.261, H.263, H.264), протоколы сигнализации (SIP, SS7, H.245, MGCP, Q.931), протоколы уровня доступа, демонстрирующие переход от специализированных и проприетарных протоколов сетей коммутации каналов к конвергентным и открытым универсальным протоколам сетей коммутации пакетов (Wi-Fi, WiMAX, xDSL).

Тема 10. Заключение.

В заключение слушателям дается целостная картина построения и взаимодействия различных сетевых технологий. Обосновывается вывод о преимуществе открытого подхода перед проприетарными решениями и о неизбежной конвергенции разных сетевых технологий.

Темы лабораторных занятий

Для выполнения лабораторных работ предполагается использовать следующие программы и оборудование: точка беспроводного доступа, эмулятор виртуальных машин Virtualbox, симулятор маршрутизатора Cisco 7200, клиентские компьютеры должны быть оборудованы адаптерами Ethernet и wi-fi, также может потребоваться наличие выделенного сервера.

Тема 1. Настройка клиентского сетевого соединения.

Данная лабораторная работа должна продемонстрировать настройку сетевого соединения локальных сетей для разных операционных систем. Настройка производится с помощью графических инструментов и внесения соответствующих изменений в конфигурационные файлы операционной системы. Учащиеся на практике осваивают следующие понятия: IP-адрес, MAC-адрес, маска сети, широковещательный адрес.

Тема 2. Настройка VPN-подключения.

Данная лабораторная работа должна продемонстрировать настройку VPN соединения локальных сетей для разных операционных систем. Настройка производится с помощью графических инструментов и внесения соответствующих изменений в конфигурационные файлы операционной системы. Учащиеся на практике осваивают следующие понятия: VPN, инкапсуляция трафика, виртуальный канал, маршрутизация, агрегирование сетей, PPTP, PPoE, PPP. Актуальность данной темы обусловлена широким применением

данных протоколов при подключении индивидуальных пользователей и домовых сетей.

Тема 3. Настройка клиента беспроводной сети Wi-Fi.

Проводится обучение настройке клиентского рабочего места к беспроводной системе в различных операционных системах с помощью графических инструментов и внесения соответствующих изменений в конфигурационные файлы. Учащимся на практике демонстрируются различные способы подключения к беспроводным сетям, в частности, подключение к точке доступа и Ad-hock-подключение.

Тема 4. Настройка клиентского коммутационного оборудования.

Изучаются устройства и принципы настройки клиентского коммутационного оборудования (DSL-модемы, аппаратные маршрутизаторы, коммутаторы, беспроводные точки доступа). Учащиеся на практике осваивают следующие понятия: NAT, VLAN, маршрутизатор, точка доступа.

Тема 5. Анализ трафика.

Изучается возможность захвата и анализа трафика с помощью программы Waresharke, настройка оборудования (сетевой адаптер, коммутатор) для захвата трафика. С помощью графической утилиты Waresharke изучается структура пакетов, наглядно демонстрируется TCP-сессия. Кроме того, рассматривается инструмент tcpdump, позволяющий просматривать трафик на более высоком уровне абстракции.

Тема 6. Настройка маршрутизации в IP-сетях.

Производится настройка статической маршрутизации на базе операционной системы Linux и коммутатора Cisco. Приводится общая концепция проектирования сети для динамической маршрутизации, настраивается динамическая маршрутизации на базе операционной системы Linux и комму-

татора Cisco. Настройка маршрутизатора Cisco демонстрируется при помощи программного симулятора маршрутизатора Cisco 7200.

Тема 7. Высокоуровневые протоколы стека TCP/IP.

С помощью текстовых команд слушатели учатся управлять с терминала telnet протоколами SMTP, POP3, IMAP. Цель данной лабораторной работы – помочь слушателям получить представление о внутреннем устройстве протоколов прикладного уровня стека TCP/IP.

Тема 8. Проектирование сетей.

Демонстрируется практическое построение сетей разных размеров. В качестве вспомогательного инструмента предполагается использование сайта www.netwizard.ru, выполненного в форме экспертной системы, позволяющей в доступной форме проектировать сети в соответствии с поставленными задачами.

Требования к контролю знаний

В процессе чтения курса предусмотрены два промежуточных контроля знаний и итоговый контроль знаний. Оценка знаний студента по каждому виду контроля осуществляется в соответствии с методикой оценки знаний.

Промежуточный контроль знаний № 1.

Контроль уровня знаний осуществляется в виде письменной контрольной работы.

Примерный перечень вопросов:

1. Структура и основные аспекты деятельности стандартизирующих организаций.
2. Физический уровень модели ISO/OSI. Среда передачи. СКС.
3. Протоколы множественного доступа. Семейство протоколов ALOHA. Рассчитать КПД.

4. Протоколы множественного доступа. Семейство протоколов CSMA.
5. Структура фрейма LLC. Применение процедур LLC.
6. Адресация MAC-уровня. Дать определения следующих понятий: глобально администрируемый, локально администрируемый, индивидуальный адрес, групповой адрес. Определить тип MAC-адреса Ethernet: A0:70:C7:F8:D0:5A.
7. Технологии Token Bus и Token Ring.
8. Адаптивные, кольцевые, высокоскоростные сети IEEE 802.17.
9. Технология Ethernet. Спецификация физической среды. Виды кадров Ethernet.
10. Технологии Fast Ethernet и Gigabit Ethernet. Отличие от Ethernet. Спецификация физической среды. Проблемы и ограничения Fast Ethernet и Gigabit Ethernet.
11. Обзор некоторых стандартов серии IEEE 802:
    1 стандарт на локальную сеть с интеграцией услуг (Integrated Services LAN) для подключения локальных сетей 802.x к общедоступным и частным магистральным сетям, таким как FDDI и ISDN (IEEE 802.9);
    2 Wireless Networks (IEEE 802.11);
    3 стандарт широкополосной беспроводной связи IEEE 802.16.

Промежуточный контроль знаний № 2.

1. IP-адресация. Разбить сеть 205.13.64.0/24 на 3 подсети, указать маску, broadcast, количество хостов.
2. Протоколы динамической маршрутизации. Рассмотреть один из протоколов: RIP, OSPF или BGP.

3. Протоколы статической маршрутизации (iproute2, click).
4. Протокол ICMP: назначение, формат ICMP-пакета. Утилита ping. Утилита traceroute.
5. Обеспечение качества обслуживания в IP-сетях.
6. Протоколы транспортного уровня: TCP и UDP.
7. Коммутация в ATM и MPLS.
8. Качество обслуживания в IP-сетях. Технологии DiffServ и IntServ.
9. Качество обслуживания в IP-сетях. Инструменты классификации и маркировки пакетов (CoS, ToS, DSCP, MPLS EXP), механизмы планирования (WFQ, RED, LLQ).
10. Протокол DNS. Адресация ENUM.

Кроме того, контроль уровня знаний включает в себя результаты защиты рефератов по тематике содержания курсов. Написание рефератов осуществляется во время самостоятельных занятий. Лучшие рефераты представляются студентами в виде презентаций и обсуждаются на лабораторных занятиях.

Примерные темы рефератов для самостоятельных занятий:

Тема 1. Технология Wi-Fi.

В реферате должен быть дан обзор стандартов технологии Wi-Fi, рассмотрена область применимости, приведена номенклатура производимого клиентского и коммутационного оборудования Wi-Fi.

Тема 2. Технология WiMAX.

В реферате должен быть дан обзор стандартов технологии WiMAX, рассмотрена область применимости, перспективы развития технологии WiMAX.

Тема 3. Технология ISDN.

В реферате должен быть дан обзор стандартов технологии ISDN, отражено современное состояние технологии ISDN. Кроме того, необходимо ответить на вопрос «что позаимствовали из технологии ISDN другие технологии?».

Тема 4. Технология ATM.

В реферате должен быть дан обзор стандартов технологии ATM, проанализированы причины коммерческой неудачи ATM. Кроме того, необходимо ответить на вопрос «что позаимствовали из технологии ATM другие технологии?».

Тема 5. Технология GPRS.

В реферате должен быть дан обзор стандартов технологии GPRS, указаны причины возникновения и коммерческого успеха технологии GPRS, проанализированы причины отказа от GPRS и перехода на другие более современные технологии.

Тема 6. Статическая маршрутизация.

В реферате должны быть даны общие понятия маршрутизации, указаны области применения статической маршрутизации, описаны возможности расширенного управления IP-трафиком посредством статической маршрутизации. Кроме того, должны быть приведены примеры на базе Cisco, Linux (iproute2, click).

Тема 7. Динамическая маршрутизация.

В реферате должны быть даны общие понятия маршрутизации, указаны области применения динамической маршрутизации, классифицированы алгоритмы динамической маршрутизации. Кроме того, должны быть приведены конкретные примеры протоколов для каждого алгоритма динами-

ческой маршрутизации, их достоинства и недостатки, примеры на базе Cisco.

Тема 8. Коммутация в ATM и MPLS.

В реферате должны быть отражены причины возникновения технологий ATM и MPLS, области их применимости, приведены примеры удачного и неудачного внедрения данных технологий.

Тема 9. Softswitch и IMS.

В реферате должен быть дан обзор стандартов и протоколов технологий Softswitch и IMS, отражено различие в концепциях.

Тема 10. Основные протоколы стека H.323.

В реферате должен быть дан обзор стандартов, основных протоколов стека H.323, сферы применения. Кроме того, должен быть описан механизм соединения в рамках стека H.323, проведено сравнение с другими стеками протоколов.

Итоговый контроль знаний.

Контроль уровня знаний осуществляется в виде письменной контрольной работы.

Примерный перечень вопросов:

1. Общий обзор технологий канального уровня для построения локальных сетей.
2. Общий обзор технологий построения глобальных и городских сетей.
3. Технология Ethernet как наиболее распространенная технология построения сетей.
4. Протокол IP как основной протокол интернет. Адресация IPv4.
5. Протокол IPv6 как эволюционное развитие протокола IPv4.

6. Недостатки протокола TCP, его сравнение с протоколом SCTP.
7. Протокол DNS. Разрешение адресов IPv4, IPv6, сервисные записи DNS, адресация ENUM.
8. Обеспечение качества обслуживания в IP-сетях.
9. MPLS: история развития, связь с ATM. Сравнение сферы применения MPLS и классической маршрутизации.
10. Общая концепция NGN. Протокол SIP.
11. Сравнение идеологий Softswitch и IMS.
11. Основные протоколы стека H.323.
12. Схема взаимодействия по протоколу H.323.
13. Схема взаимодействия по протоколу SIP.
14. Возможные схемы взаимодействия IP-сетей и ТфОП.

Литература

Обязательная

1. *Таненбаум Э. Компьютерные сети (3 или 4 изд.) // Спб.: Изд-во «Питер», 2003.*
2. *В. Г. Олифер, Н. А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. — Учебник для вузов. 3-е изд. — СПб.: Питер, 2006. — 958 с.*
3. *Семенов Ю.А. Протоколы Internet. — Изд-во «Горячая линия–Телеком». — 2005.*
4. *Новиков Ю.В., Кондратенко С.В. Основы локальных сетей. — Интернет-университет информационных технологий. — ИНТУИТ.ру, 2005.*
5. *Гольдштейн А.Б., Гольдштейн Б.С. SOFTSWITCH. — СПб.: БХВ . — Санкт-Петербург, 2006. — 368 с.*

6. *Гольдштейн Б.С. Интеллектуальные сети. — Издательство: Радио и связь, 2000.*

Дополнительная литература и источники Интернет

1. *Самуйлов К.Е, Кулябов Д.С. «Сети и системы телекоммуникаций». Учебно-методическое пособие. // М.: Изд-во РУДН, 2002.*
2. *Сунчелей И.Р., Стрижаков С.К., Семенов А.Б. Структурированные кабельные системы. — 5-е изд. — Издательство: Компания АйТи, ДМК. 2004. — 640 с.*
3. *Рошан П., Лиэри Дж. Основы построения беспроводных локальных сетей стандарта 802.11. — М.: Издательский дом "Вильямс", 2004. — 304 с.*
4. *Вишневский В., Ляхов А., Портной С, Шахнович И. Широкополосные беспроводные сети передачи информации. — М.: Эко-Трендз, 2005. — 592 с.*
5. *Гулевич Д. С. Сети связи следующего поколения. — БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий. — ИНТУИТ.ру, 2007.*
6. *Duglas E. Comer Internetworking with TCP/IP: Principles, Protocols, and Architecture. — Prentice Hall, 1995.*
7. *Craig Hunt TCP/IP Network Administration. — 2 ed. — O'Reilly & Associates, 1998.*

Аннотированное содержание курса.

Первый модуль трудоемкостью в 1 кредит составляют:

- теоретический материал, излагаемый в темах 1 – 2 программы курса (1-5 лекции календарного плана курса);
- содержание семинарских занятий в течение 10 академических часов;

В конце модуля проводится промежуточный контроль знаний № 1.

Второй модуль трудоемкостью 2 кредита составляют:

- теоретический материал, излагаемый в темах 3-6 программы курса (в лекциях 7 – 13 календарного плана курса);
- отработка практических заданий в виде рефератов в течении 20 часов самостоятельных занятий;
- содержание семинарских занятий в течение 14 академических часов.

В конце модуля проводится промежуточный контроль знаний № 2.

Третий модуль трудоемкостью в 1 кредит составляют:

- теоретический материал, излагаемый в темах 5-9 программы курса (в лекциях 13 – 19 календарного плана курса);
- отработка практических заданий в виде рефератов в течении 14 часов самостоятельных занятий;
- содержание семинарских занятий в течение 10 академических часов.

В конце модуля проводится итоговый контроль знаний.

Календарный план курса

	Виды и содержание учебных занятий			
Неде-ля	**Лекции**	**Число часов**	**Лабораторные занятия**	**Число часов**
1	Базовые понятия в области систем и сетей телекоммуникаций Общая характеристика проблемной области. Стандартизирующие организации. Общие принципы построения модели взаимодействия открытых систем (ISO/OSI), иерархия протоколов различных стеков протоколов.	**2**	Ознакомление с основным сетевым оборудованием, СКС, устройством сети.	**2**
2	Физический уровень модели ISO/OSI. Обзор возможных сред передачи, СКС, методы кодирования сигнала и сферы их применения.	**2**	Настройка клиентского коммутационного оборудования.	**2**
3,4,5	Канальный уровень модели ISO/OSI. Методы и протоколы (семейства ALOHA и CSMA) доступа к среде. Технологии сетей (Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring,	**6**	Настройка клиента беспроводной сети Wi-Fi. Настройка маршрутизатора Wi-Fi. Развертывание ad-hock сети.	**6**

	Виды и содержание учебных занятий			
Неде-ля	**Лекции**	**Число часов**	**Лабораторные занятия**	**Число часов**
	100VG–AnyLAN, Wireless Networks, WiMAX, Bluetooth, FDDI, Frame Relay).			
6	**Промежуточный контроль знаний № 1**			**2**
7	Сетевой уровень модели ISO/OSI. Формат кадра IP, IP-адресация (IPv4 и IPv6), взаимодействие межсетевого уровня с физическим.	**2**	Настройка клиентского сетевого соединения.	**2**
8	Сетевой уровень модели ISO/OSI. Классификация алгоритмов маршрутизации и, собственно, протоколы статической (iproute2, click) и динамической маршрутизации (RIP, OSPF, BGP), сфера их применения, достоинства и недостатки.	**2**	Настройка VPN-подключения.	**2**
9	Сетевой уровень модели ISO/OSI. Протоколы межсетевого уровня стека протоколов TCP/IP (ARP, RARP, ICMP), их назначение.	**2**	Настройка маршрутизации в IP-сетях.	**2**

	Виды и содержание учебных занятий			
Неде-ля	**Лекции**	**Число часов**	**Лабораторные занятия**	**Число часов**
10	Транспортный уровень модели ISO/OSI. Протоколы транспортного уровня стека протоколов TCP/IP: TCP (формат TCP-пакета, алгоритм установления связи (сессия TCP), параметры передачи, MTU, надежная доставка, технология скользящего окна, процедура «медленный старт»), UDP (формат UDP-пакета, псевдозаголовок, ненадежная передача),	**2**	Анализ трафика.	**2**
11	Транспортный уровень модели ISO/OSI. Протоколы транспортного уровня стека протоколов TCP/IP: DCCP, SCTP.	**2**	Анализ трафика.	**2**
12	QoS и передача мультимедийных данных. ATM, MPLS, DiffServ, IntServ. Инструменты классификации и маркировки пакетов (CoS, ToS, DSCP, MPLS EXP),	**2**	Настройка управления трафиком на Linux-маршрутизаторе и маршрутизаторе Cisco.	**2**

	Виды и содержание учебных занятий			
Неде-ля	**Лекции**	**Число часов**	**Лабораторные занятия**	**Число часов**
	механизмы планирования (WFQ, RED, LLQ и др.) и механизмы канального уровня (policing – ограничение скорости и shaping – выравнивание трафика, фрагментация и чередование пакетов, компрессия).			
13	Некоторые протоколы верхних уровней (DNS, ENUM)	**2**	Высокоуровневые протоколы стека TCP/IP.	**2**
14	**Промежуточный контроль знаний № 2**			**2**
15, 16	Некоторые мультисервисные сети и системы связи (ISDN, H.323)	**4**	Проектирование сетей.	**4**
17, 18	Основные подходы к построению сетей следующего поколения (NGN): Softswitch и IMS. Технологии построения сетей следующего поколения (NGN). Основные группы протоколов: протоколы кодирования (G.711, G.723.1, G.728, G.729, H.261, H.263, H.264),	**4**	Проектирование сетей.	**4**

	Виды и содержание учебных занятий			
Неде-ля	**Лекции**	**Число часов**	**Лабораторные занятия**	**Число часов**
	протоколы сигнализации (SIP, SS7, H.245, MGCP, Q.931).			
19	Обзорная лекция. Подготовка к итоговому контролю знаний.	**2**	Подготовка к итого-вому контролю зна-ний.	**2**
20	**Итоговый контроль знаний**			**2**

4. Описание системы контроля знаний

Шкала балльно-рейтинговой системы.

Баллы за семестр	Автоматическая оценка		Баллы за итоговый контроль знаний	Общая сумма баллов	Итоговая оценка
	Итоговая оценка	Дополнительные баллы			
78 – 80	5	по 5 баллов за каждый свыше 76**	0 – 20*	86 – 100	5
69 – 77	4	Нет	0 – 20* 0 – 20*	86 – 97 69 – 85	5 4
51 – 68	Нет	Нет	0 – 20 0 – 20 0 – 20	86 – 88 69 – 85 51 – 68	5 4 3
41 – 50	Нет	Нет	0 – 20 0 – 20 0 – 20	69 – 70 51 – 68 41 – 50	4 3 2
< 41	2	Нет	Нет	Нет	2

Примечания к таблице шкалы балльно-рейтинговой системы.

* студент имеет право не проходить итоговый контроль знаний

** дополнительные баллы начисляются автоматически: за 78 баллов, набранных в семестре, начисляется дополнительно 10 баллов (общая сумма баллов – 88); за 79 баллов – 15 баллов (94); за 80 баллов – 20 баллов (100).

Шкала диапазонов итоговой оценки по 5-и балльной системе выбрана следующим образом.

100 – балльная система	Итоговая оценка
86 – 100	5
69 – 85	4
51 – 68	3
0 – 50	2

Порядок начисления баллов

1. Порядок начисления баллов за семестр.

1.1 Общая оценка работы в семестре. Посещаемость занятий, активность работы на семинарских занятиях: 0 – 10 баллов

1.2 Промежуточный контроль знаний № 1: 0 – 20 баллов

Вопрос 1: 0 – 10 баллов

Вопрос 2: 0 – 10 баллов

1.3 Промежуточный контроль знаний № 2: 0 – 25 баллов

Вопрос 1: 0 – 5 баллов

Вопрос 2: 0 – 10 баллов

Вопрос 3: 0 – 10 баллов

1.4 Реферат: 0 – 25 баллов

Содержание реферата: 0 – 15 баллов

Качество презентации реферата: 0 – 10 баллов

2. Порядок начисления баллов за итоговый контроль знаний.

2.1 Контрольная работа № 2: 0 – 20 баллов

Вопрос 1: 0 – 10 баллов

Вопрос 2: 0 – 10 баллов

Пример применения методики оценки знаний

1. Начисление баллов за семестр.

1.1 Студент посетил не менее 95% занятий. На семинарских занятиях не менее 2-х раз принимал участие в обсуждениях, правильно и четко формулировал свои мысли, использовал правильную терминологию и показал умение работать с рекомендованной литературой.

Набранные баллы: 10 баллов.

1.2 На контрольной работе (промежуточный контроль знаний № 1) студент письменно отвечал на следующие вопросы:

Вопрос 1. Физический уровень модели ISO/OSI. Среда передачи. СКС..

В ответе на вопрос были охарактеризованы не все услуги физического уровня модели ISO/OSI – отсутствует услуга идентификации канала и услуга оповещения об ошибках. При этом в ответе на вопрос дана классификация сред передачи данных и перечислены их конкретные реализации, раскрыто понятие СКС, но не приведены конкретные примеры СКС.

Набранные баллы: 8 баллов.

Вопрос 2. Протоколы множественного доступа. Семейство протоколов ALOHA. Рассчитать КПД.

В ответе на вопрос дана классификация моделей доступа к среде, приведены примеры, подробно рассмотрена схема работы протоколов семейства ALOHA, но допущена грубая ошибка при расчете КПД.

Набранные баллы: 7 баллов.

1.3 На промежуточном контроле знаний № 2 студент письменно отвечал на следующие вопросы:

Вопрос 1. Протокол ICMP: назначение, формат ICMP-пакета. Утилита ping. Утилита traceroute.

В ответе на вопрос протокол ICMP был полностью охарактеризован, были допущены незначительные ошибки при описании формата ICMP-пакета, на примерах объяснено назначение и результат запуска утилит ping и traceroute.

Набранные баллы: *5 баллов.*

Вопрос 2. Протоколы маршрутизации. Рассмотреть протокол OSPF.

В ответе на вопрос дана классификация протоколов маршрутизации, указано преимущество протокола OSPF перед протоколом RIP, но отсутствует понимание процедуры агрегирования адресов.

Набранные баллы: *7 балла.*

Вопрос 3. Качество обслуживания в IP-сетях. Технологии DiffServ и IntServ.

Ответ на вопрос был исчерпывающем без замечаний.

Набранные баллы: *10 баллов.*

1.4 Тема реферата: *Softswitch и IMS.*

При написании реферата студент помимо рекомендованной литературы самостоятельно подобрал дополнительные источники информации в Интернет. Объем реферата составил 30 страниц с рисунками и диаграммами, реферат оформлен в соответствии с требованиями написания учебно-научных материалов. При написании реферата студент активно использовал возможности виртуального кабинета преподавателя, задавал вопросы, выкладывал промежуточные версии реферата. В реферате студент дал обзор стандартов NGN, провел сравнительный анализ концепций Softswitch и IMS, описал области практического применения каждого подхода и указал перспективы развития каждого подхода. Допустил незначительные неточности при описании архитектуры, сделал несколько опечаток и не полностью заполнил список аббревиатур и терминов.

Набранные баллы: *14 баллов.*

Студент подготовил в электронном виде презентацию по содержанию реферата, сделал 15-и минутный доклад, четко отвечал на вопросы преподавателя и других слушателей.

Набранные баллы: *10 баллов.*

2. Начисление баллов за итоговый контроль знаний.

2.1 На контрольной работе (итоговый контроль знаний) студент письменно отвечал на следующие вопросы:

Вопрос 1. Общий обзор технологий канального уровня для построения локальных сетей.

Ответ на вопрос был исчерпывающем без замечаний.

Набранные баллы: *10 баллов.*

Вопрос 2. Обеспечение качества обслуживания в IP-сетях.

Ответ на вопрос был исчерпывающем без замечаний.

Набранные баллы: *10 баллов.*

Таким образом, в течение семестра студент набрал следующие баллы.

Посещаемость занятий и активность: *10 баллов*

Промежуточный контроль знаний № 1: *15 баллов*

Промежуточный контроль знаний № 2: 23 *баллов*

Реферат и презентация: 24 *балла*

Итого в семестре *N* =: 72 *балла*

Для оценки работы в семестре применяется вторая строка шкалы балльно-рейтинговой системы, поскольку $69 < N < 77$.

Итоговая оценка за работу в семестре по 5 балльной шкале: *4 (хорошо).*

Студент имеет право получить автоматическую оценку и не проходить итоговый контроль знаний (примечание *).

Студент для повышения оценки прошел итоговый контроль знаний.

Итоговый контроль знаний *M* = : 20 *баллов*

Общая сумма баллов N + M =: *72 + 20 = 92 балла*

Итоговая оценка по 5 балльной шкале: *5 (отлично).*

Академическая этика, соблюдение авторских прав.

Во всех компонентах УМК ссылки на литературные источники и источники Интернет являются актуальными, тщательно выверенными и снабженными «адресами». В тексты не включены выдержки из работ других авторов без ссылки на соответствующий источник, не пересказаны работы других авторов близко к их тексту и без ссылки на соответствующий источник. В УМК не использованы чужие идеи без указания первоисточников. Это распространяется на литературные источники (монографии, учебники, статьи и пр.) и источники Интернет, для которых в необходимых случаях указан полный адрес соответствующего сайта.